

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S)			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b. OFFICE SYMBOL (If applicable) Code 37	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School			
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000		7b. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000			
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER			
8c. ADDRESS (City, State, and ZIP Code)		10. SOURCE OF FUNDING NUMBERS			
		Program Element No.	Project No.	Task No.	Work Unit Accession Number
11. TITLE (Include Security Classification) DESIGN AND IMPLEMENTATION OF A PATIENT TRACKING AND RECALL SYSTEM FOR BRANCH DENTAL CLINIC MONTEREY					
12. PERSONAL AUTHOR(S) Steele, Timothy P					
13a. TYPE OF REPORT Master's Thesis	13b. TIME COVERED From To	14. DATE OF REPORT (year, month, day) 1992, March	15. PAGE COUNT 127		
16. SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.					
17. COSATI CODES			18. SUBJECT TERMS (continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUBGROUP	Information System Analysis; Software Development; Patient Tracking and Recall; Relational Database		
19. ABSTRACT (continue on reverse if necessary and identify by block number) This thesis analyses the information system requirements of Branch Dental Clinic, Monterey, and develops a computer application to automate the clinic's patient tracking and recall process. The application replaces an existing mainframe-based, single-file system with a PC-based, relational database management system that provides greater functionality, enables increased productivity, improves data integrity and accuracy, and includes currently lacking security features and administrative functions.					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS REPORT <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL Hemant K. Bhargava			22b. TELEPHONE (Include Area code) (408) 646-2264		22c. OFFICE SYMBOL Code AS/BI

Approved for public release: distribution is unlimited

Design and Implementation of a Patient Tracking and Recall System
for Branch Dental Clinic Monterey

by

Timothy P. Steele
Lieutenant Commander, United States Navy
B.A., University of Washington, 1979

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
March 1992

ABSTRACT

This thesis analyzes the information system requirements of Branch Dental Clinic, Monterey, and develops a computer application to automate the clinic's patient tracking and recall process. The application replaces an existing mainframe-based, single-file system with a PC-based, relational database management system that provides greater functionality, enables increased productivity, improves data integrity and accuracy, and includes currently lacking security features and administrative functions.

367 915
2.1

TABLE OF CONTENTS

I. INTRODUCTION AND PRELIMINARY INVESTIGATION	1
A. BACKGROUND	1
B. PROBLEM DEFINITION	3
C. SCOPE	5
D. EVALUATION OF ALTERNATIVE SOLUTIONS	6
1. Cost Feasibility	6
2. Technical Feasibility	7
3. Schedule Feasibility	8
II. REQUIREMENTS ANALYSIS	9
A. DATA REQUIREMENTS	9
1. Object Development	9
2. Domain Definition	11
B. APPLICATION REQUIREMENTS	11
1. Processes	11
2. Operational and Administrative Requirements	13
3. Environmental Requirements	14
III. SYSTEM DESIGN	15
A. LOGICAL DATABASE DESIGN	15
1. Object to Relation Transformations	15

2. Relation Descriptions	16
B. APPLICATION DESIGN	17
IV. SYSTEM IMPLEMENTATION	21
V. SUMMARY AND RECOMMENDATIONS	22
A. SUMMARY	22
B. RECOMMENDATIONS	22
LIST OF REFERENCES	24
APPENDIX A: OBJECT SPECIFICATIONS	25
APPENDIX B: UPDATE, DISPLAY, AND CONTROL MECHANISMS	28
APPENDIX C: USER'S MANUAL	34
APPENDIX D: RELATION DEFINITIONS	35
APPENDIX E: PROGRAM CODE	36
INITIAL DISTRIBUTION LIST	37

ACKNOWLEDGEMENT

Special acknowledgement must be given to my wife, Anne, for her support and forbearance as I spent long hours at the computer, seemingly oblivious to the meaning of life, working on this thesis. I am truly in her debt.

I. INTRODUCTION AND PRELIMINARY INVESTIGATION

This thesis designs, documents, and implements a computer application to perform dental patient tracking and recall functions for the Branch Dental Clinic, Monterey (BDCM). Information that was collected during a preliminary investigation of the information system requirements of BDCM is presented in this chapter. Specifically, the relevant background of BDCM and the information system problems that led to the conduct of the thesis are presented, the scope of the project is defined, and three alternative solutions are evaluated.

A. BACKGROUND

BDCM provides regular dental care and emergency dental treatment to all active duty military staff and students stationed both at the Naval Postgraduate School (NPS) and the various NPS tenant commands. Dental appointments are regularly scheduled based on a four-class rating system (1 to 4, in order of increasing priority) indicating the member's need for treatment. Emergency care is provided whenever required.

Interviews with the BDCM Director and staff identified four major information-oriented activities within the clinic: (1) appointment scheduling, (2) inventory management, (3) maintenance of a Dental Information and Retrieval System (DIRS) as prescribed by higher authority, and (4) patient tracking and recalls. With regard to appointment scheduling and inventory management activities, BDCM satisfaction with

current manual methods was found to be high. Moreover, the clinic Director felt strongly that attempts to computerize these two functions, given the relatively low volume of activity, would not increase efficiency or effectiveness. Hence, these two business functions were dropped from further investigation.

The DIRS system operates on a personal computer (PC) and consists of proprietary software provided by the Navy Regional Dental Center (NRDC) for use at all subordinate branch clinics. Since NRDC mandates that branch clinics use DIRS to collect and report detailed data on all dental care provided, further analysis of this activity was unnecessary.

Patient tracking and recall functions at BDCM are partially automated by a mainframe-based, single-user, single-file database management system. It is this system and the requirements of the patient tracking and recall process that the remainder of this thesis addresses.

The mainframe-based database application allows data entry and updating, tracks members' dental health status (class), generates recall notices, prints sorted member rosters, and provides operational readiness summary statistics. When members check their records into the clinic a dentist's review of their dental records results in a class rating being assigned. A class rating of "1" indicates no need for dental treatment beyond a mandatory annual examination (a T2-exam). A class rating of "2" or "3" indicates a need for additional treatment. A class rating of "4" indicates the member is past due for an annual exam (it is assigned regardless of dental health). Just prior to a member's T2-exam anniversary, he/she is notified by memorandum to make an

appointment for an annual exam using an automated patient recall system. Computer generated recall letters are routed to student mail center (SMC) mailboxes or staff offices as appropriate.

B. PROBLEM DEFINITION

The existing application for patient recalls was written several years ago for use on the NPS mainframe computer. When the system was installed it provided significant benefits over the previous labor and time intensive manual recall process. However, the system was crude in its interface, limited in functionality, and difficult to use. Moreover, due to turnover of personnel since its installation, none of the current staff are familiar with the history of the system; no documentation can be found; and no system maintenance is available.

Interviews with end-users revealed five general problem areas with the mainframe-based system: poor access and responsiveness, unfriendly user-interface, inadequate data validation checks, absence of documentation, and incomplete functionality. Examples of specific problems highlighted by end-users in each of these general areas are presented below.

Limited mainframe access and poor responsiveness have been longstanding limitations. BDCM access to the mainframe is via communications software and 1200 baud modem from the clinic PC. By today's standards, this data transfer rate is slow. The system frequently responds slowly during working hours due to both the high number of users and resource-intensive computing tasks. Heavy use of the mainframe

by modem users combined with the limited number of modem receiving lines (16 at the time of this investigation) results in the frequent inability to access the system as needed. This necessitates periodic off-hour work by BDCM staff and delays response to telephone queries from NRDC regarding operational readiness.

Unfriendliness of the user-interface is a significant problem, particularly for new users. In most instances the user is presented with only a blank screen and a prompt, which specifies which application module is active (e.g., main, add, edit, delete, print). A rudimentary help function, when invoked, provides a list of options for the active module. Hence, unless all commands are memorized, the user must continuously invoke the help function to navigate and use the system. Data entry itself is facilitated somewhat by a field list from which the user selects a field to enter or edit, but it remains a cumbersome process. The user must select a field from the list, enter the data, and select another field from the list rather than simply automatically moving from one field to the next. Additionally, during record appending the field listing scrolls up and off the screen, leaving no hint of the remaining fields that require additional data entry.

The inadequacy of field validation checks in the mainframe application has allowed a cumulative deterioration in the accuracy and completeness of records in the database. For example, numbers are improperly allowed in various name fields. Moreover, since member records are indexed by name rather than Social Security Number (SSN), two people with the same name are prohibited from being entered properly into the database. In such instances, the user must deliberately attempt to circumvent or "trick" the system by, for example, putting in a middle initial for one member but not the other. Related

to this, the system saves a new record whenever data is entered into the name field, regardless of content and regardless of whether the record has any other fields completed. Over time the database has accumulated much erroneous data and many incomplete records. Cleaning the database has been problematic since records cannot be located and edited or deleted unless an exact name match is entered.

The lack of system documentation has forced end-users to learn the system by experimentation. The total functionality of the system is not immediately obvious and can remain undiscovered and unutilized. Moreover, the logic underlying critical processes, such as the triggering of recall notices or updating dental class status remains unspecified. The lack of documentation has also precluded improving the functionality of the system and implementing fixes. For example, necessary follow-up form letters that are not included in the present system must be externally word-processed for each individual. Additionally, hard-coding of the signature name on recall letters has resulted in a long since-transferred Director's name appearing on the recalls sent to members.

C. SCOPE

The scope of this thesis is limited to the patient tracking and recall process. As noted previously, there are other business functions within the clinic, yet the patient tracking and recall process is the only information-intensive business function left up to local implementation that remains problematic.

D. EVALUATION OF ALTERNATIVE SOLUTIONS

Given that the problems with the existing patient tracking and recall system were deemed significant enough to warrant remediation, three alternative solutions were evaluated. The first alternative involved improving both the hardware and software associated with the mainframe-based system: replacing the modem connection with an on-line terminal, rewriting the software for increased functionality and ease of use, and documenting the system. The second and third alternatives involved designing and implementing a PC-based database management system to replace the existing mainframe application, the difference being whether a multi-user versus a single-user configuration should be developed. Multi-user capability was considered a "nice-to-have" feature that might be useful sometime in the future, yet it was clearly not a requirement for satisfactory performance of patient tracking and recall functions. Should a PC-based solution be selected, NRDC stipulated that it must be a compiled application that would not be subject to potential modification by inexperienced clinic staff.

1. Cost Feasibility

At the outset, NRDC made it clear that no funds were available to support improving the existing patient tracking and recall system. This limitation alone ruled-out upgrading the mainframe-based system—the cost of terminal acquisition and connection was prohibitive. Moreover, additional funds would be required to pay a technical expert to rewrite and document the mainframe software. Similarly, to exploit multi-user capability in a PC-based system would require additional funding to purchase required hardware. Hence, these two alternatives were eliminated from further consideration.

Designing and implementing a single-user, PC-based database management system was attractive from a cost standpoint. The development cost of such a system would be limited to the personal time and effort of the author. Further, appropriate development hardware (an IBM-compatible 80386 computer) and software (Foxpro 2.0 and Foxpro 2.0 Distribution Kit, a dBase-compatible development system with compiler) was already owned by the author. In addition, BDCM would not be required to purchase any additional hardware; their existing computer equipment could be used to evaluate prototypes and to install the final working system. BDCM staff were enthusiastic and committed to assisting with the development process.

2. Technical Feasibility

BDCM owned a Zenith 286 PC and peripherals that were compatible with the foreseeable processor, memory, storage, and video requirements of a new PC-based application. Moreover, Foxpro 2.0 can create applications able to run on any IBM-compatible PC with a minimum of 512K of random access memory (RAM) [Ref. 1]. Preliminary tests of routine database operations (browse, index, sort) with a test database approximately the same size as that of the existing mainframe data file (2000 records with 15 fields per record) using Foxpro 2.0 were successful on the BDCM PC and demonstrated acceptable speed of operations with only 512K of RAM.

Future maintenance of the application would not be provided by the author. Discussion of this issue with both NRDC and BDCM indicated that this was acceptable to them. It was agreed that the application should run on any minimally configured IBM-compatible computer to enable portability and that support for a standard dot-matrix

printer should be provided. Program code and documentation would be included with the delivered application to support future maintenance. (NRDC and BDCM acknowledged that any future maintenance would require purchase of Foxpro 2.0 and the Foxpro 2.0 Distribution Kit. Intermediate-level dBase or Foxpro programming skills would also be required.)

3. Schedule Feasibility

Based on the findings of the preliminary investigation, with detailed system analysis to begin 15 August, 1991, implementation of a fully operational PC-based system was scheduled for completion by 1 February, 1992. This left two months for correction of unforeseen problems before departure of the author.

II. REQUIREMENTS ANALYSIS

This chapter discusses the requirements phase of project development. The purpose of this phase of development was twofold: (1) during this phase the specific data requirements (objects) that must be represented in the database were defined and (2) the application or functional requirements which support the database were outlined.

A. DATA REQUIREMENTS

Initially, interviews were conducted with the BDCM Director and the dental staff responsible for hands-on use of the existing database. These interviews provided a general idea of the scope and objectives for an upgraded patient tracking and recall system. Working backwards from the existing application's outputs, preliminary object specifications and views were then developed and presented to the dental staff for feedback. Further discussions led to adjustments of the object specifications that satisfactorily met the clinic's needs.

1. Object Development

Important entities identified in the patient tracking and recall process are represented as the objects MEMBER, ACTIVITY, and CURRICULUM shown in Figure 1 below. Each of the objects possesses a collection of named properties. The properties listed within each diagram that are capitalized and within small boxes are themselves objects. The subscript "MV" denotes that the property is multi-valued. The MEMBER

object represents patients who have "checked-in" with the clinic upon arrival to NPS or an NPS tenant command. As can be seen in Figure 1, the ACTIVITY and CURRICULUM objects are properties of the MEMBER object. They associate each member with the properties of a specific activity and/or curriculum.

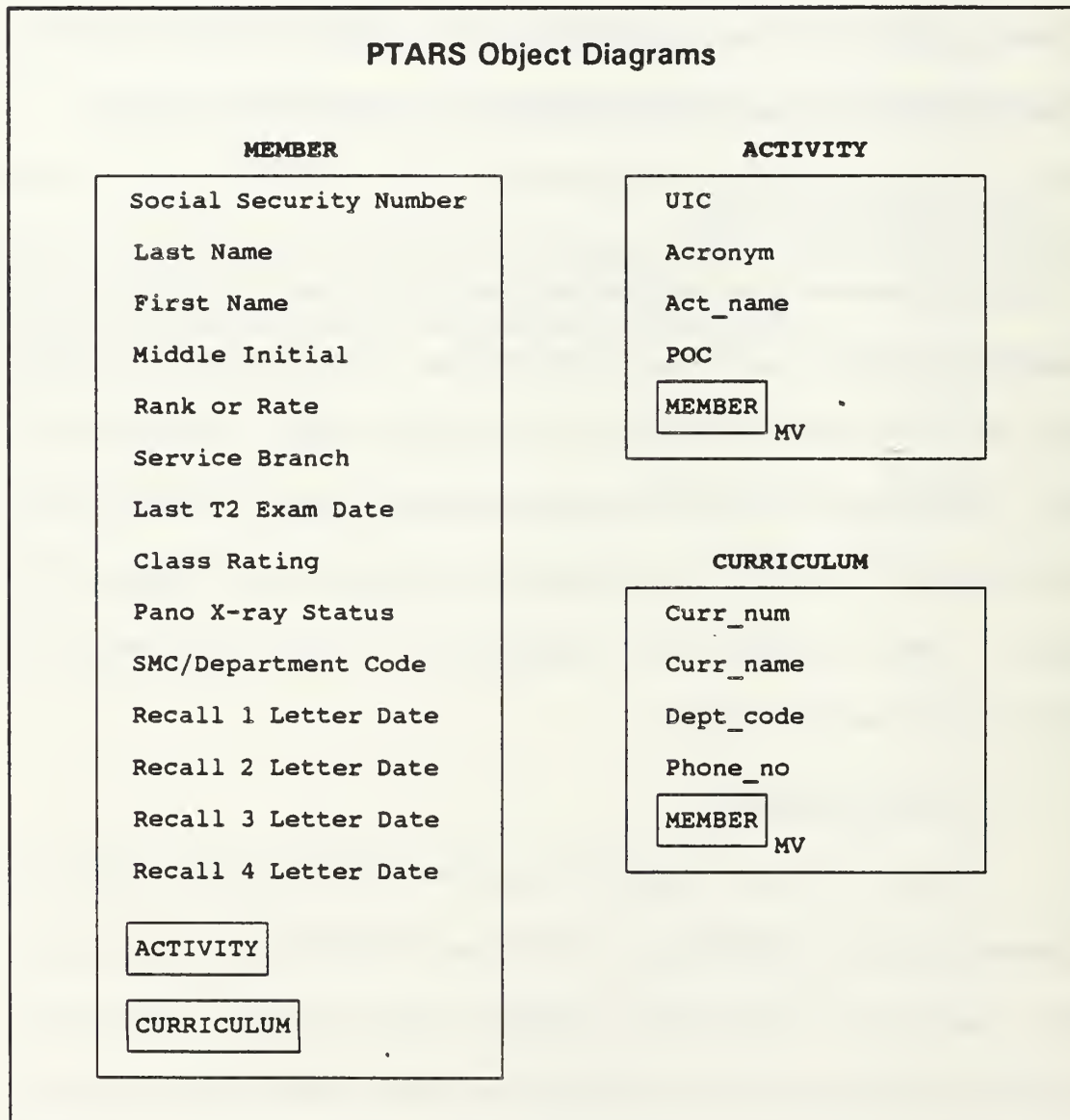


Figure 1. Object Diagrams

The ACTIVITY object represents each of the various commands served by BDCM. Note that the multi-valued MEMBER object is also a property of the ACTIVITY object. That is, a specific ACTIVITY can have multiple members.

The CURRICULUM object represents each of the many different curriculums offered at NPS. The MEMBER object is a multi-valued property of the CURRICULUM object; many students can belong to any given curriculum.

2. Domain Definition

The object diagrams were used to summarize knowledge of the objects and to present it to the users in an unambiguous fashion. Following user validation of the object representations, domain definitions were established. The domain of a property is the set of all possible values a property can have. Each domain definition contains a physical description of the type of data (e.g., numeric versus character) and any value constraints. Each definition also describes the function or purpose of the property. Refer to Appendix A for detailed object specifications, including object and domain definitions.

B. APPLICATION REQUIREMENTS

1. Processes

Building upon the data requirements discussed in the previous section, major processes within the patient tracking and recall process were identified through discussions with BDCM end-users. A level-1 data flow diagram (DFD), shown in Figure

2 below, was developed as a basis for validating analyst understanding of the processes with end-users.

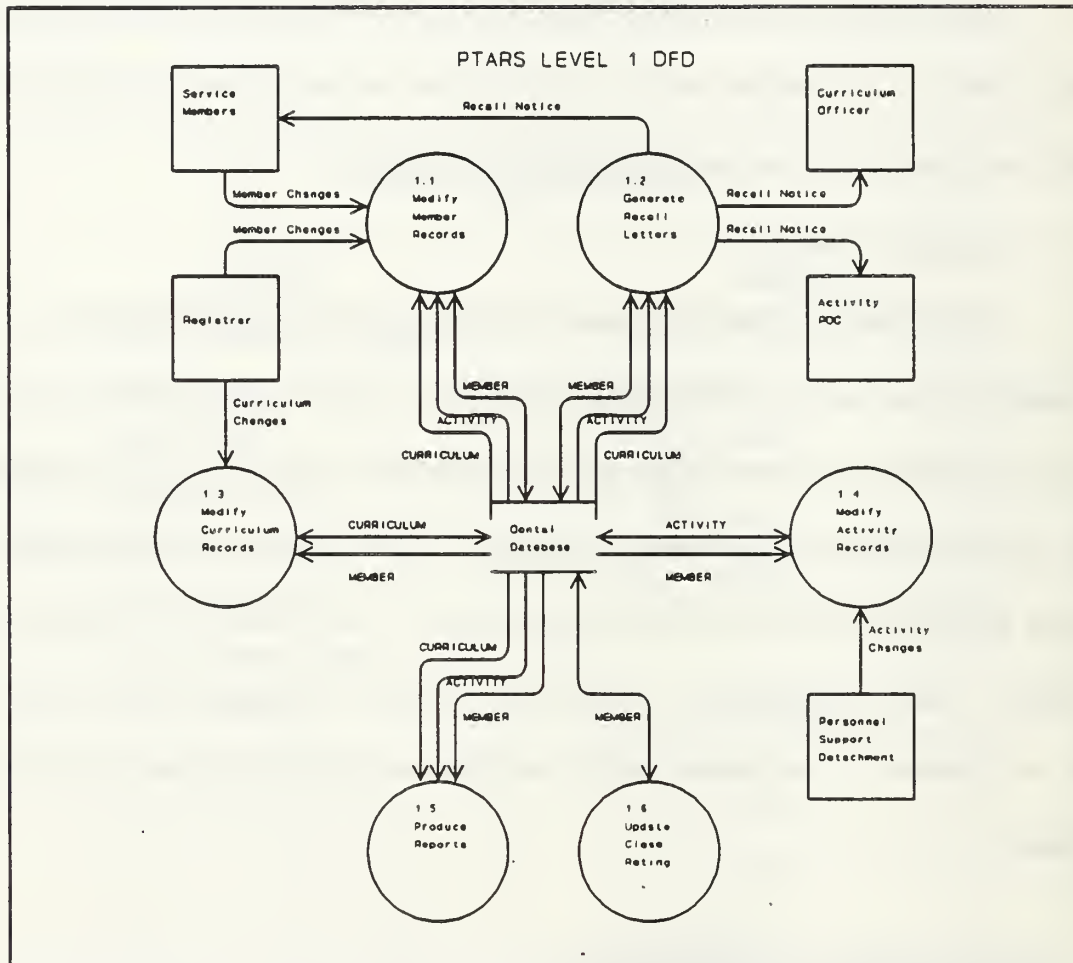


Figure 2. Level 1 Data Flow Diagram

Entities external to the system are shown in Figure 2 as square boxes and include Service Member, Registrar, Personnel Support Detachment (PSD), Curriculum Officer, and Activity Point of Contact. These entities are sources of data and/or recipients of system outputs (as indicated by the direction of the data flow arrows). The

numbered processes (denoted within the circles) summarize the operations involved in the overall patient tracking and recall process. Processes 1.1, 1.3, and 1.4 comprise the append, edit, and delete operations for the objects, MEMBER, CURRICULUM, and ACTIVITY. Process 1.2 involves the operations associated with generating and printing recall letters. An Operational Readiness report and various sorted rosters are produced in process 1.5. Member dental class is automatically updated to class 4 in process 1.6 for those individuals who have not had an annual examination within 12 months.

Following validation of the information presented by the level-1 DFD, a summary of system update, display, and control mechanisms was developed based on structured interviews with end-users. (See Appendix B.) During this process, information pertaining to each object was obtained that included inputs, outputs, processing notes, volume, and frequency. This information clarified what must be done within each object view.

Prototypes of forms, reports, recall letters, and menus were developed using Foxpro "power tools" (i.e., the Screen Builder and the Report Writer). These early prototypes clarified the expectations of end-users regarding the format of the user-interface and the display of information.

2. Operational and Administrative Requirements

System operational and administrative requirements were identified through discussions with BDCM staff. Operational requirements for the system are listed below:

- Single-user, PC-based application, operable on an "as needed" basis by the BDCM Administrative Petty Officer and/or the BDCM Receptionist

- Portable/re-installable to different, compatible PC
- Extensive "Help" available on-line
- Database backup/restore utilities
- System date and time change utilities
- System-access password protection; password change capability
- Database packing capability

Although it was agreed that program maintenance would not be possible with the compiled application, Foxpro 2.0 program code would be given to BDCM. Hence, should maintenance become critical at some point, modification of the application would be possible with the purchase of Foxpro 2.0 and the Foxpro 2.0 Distribution Kit. A User's Manual (see Appendix C) would be supplied to provide structured guidance for system use, data security and integrity, database backups and restorations, and system optimization.

3. Environmental Requirements

In an efficient system much of the member, activity, and curriculum data should be provided from a master database, shared with the Registrar and PSD. However, this is currently not possible since the data structure and hardware are not compatible. Until such time as the various NPS support entities/ADP-systems can communicate directly, it is incumbent upon the BDCM staff to take the initiative to obtain updated, hard-copy rosters from these two data sources as they become available.

III. SYSTEM DESIGN

In this chapter the two components of system design, logical database design and application design, are discussed. The objective of the design phase was to produce both the logical and physical details of the database and its application. Designing the logical database involved developing a "blueprint" of the database structure. From this blueprint a physical database was designed and the application was developed.

A. LOGICAL DATABASE DESIGN

1. Object to Relation Transformations

The design of the logical database was based on the relational database model [Ref. 2]. The objects MEMBER, ACTIVITY, and CURRICULUM, were transformed into a relational diagram. Figure 3, the relational diagram, shows the three relations that resulted: (1) the compound MEMBER object was transformed into the three relations MEMBER, ACTIVITY, and CURRICULUM; (2) the compound ACTIVITY object was transformed into the two relations MEMBER and ACTIVITY; and (3) the compound CURRICULUM object was transformed into the two relations MEMBER and CURRICULUM.

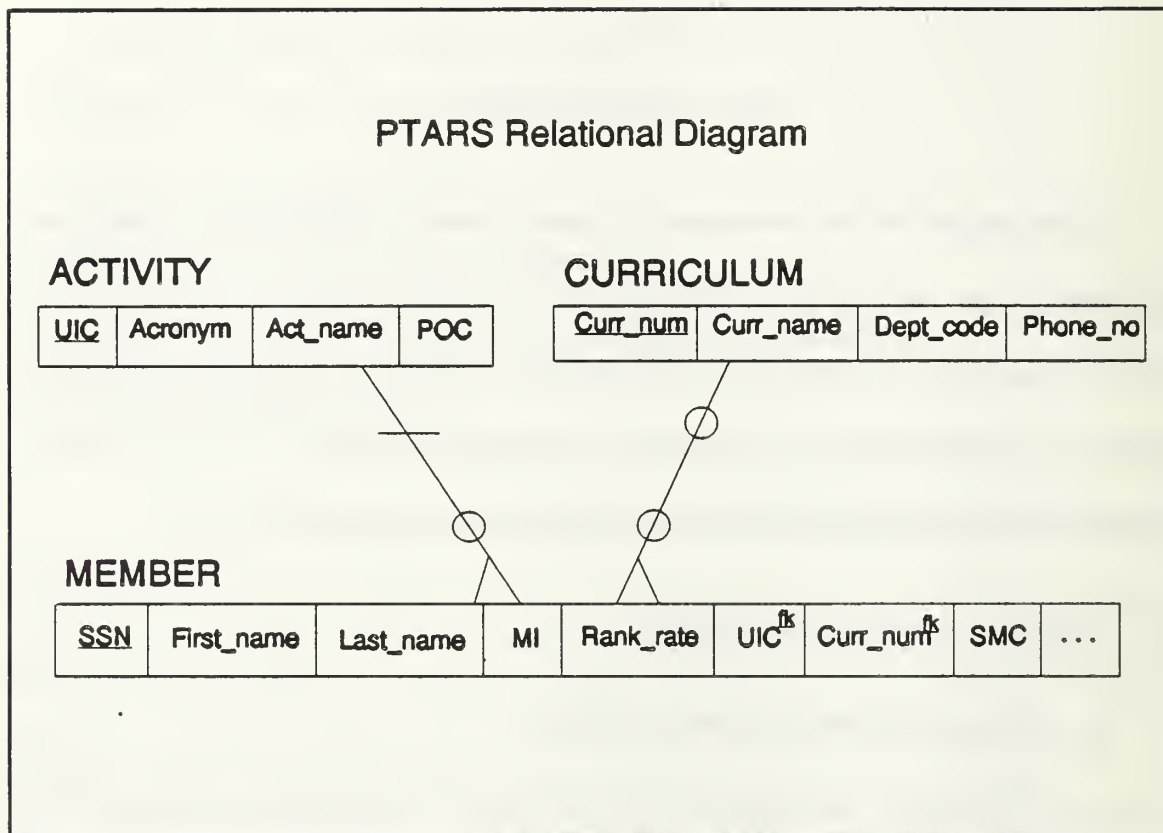


Figure 3. Relational Diagram

2. Relation Descriptions

Each of the three relations are reflections of the original objects with appropriate foreign keys included. Key data are denoted in Figure 3 by underlining. Foreign keys are denoted with the underlined superscript, ^{fk}. Summary descriptions of each of the relations are presented below. (Refer to Appendix D for detailed relation definitions.)

MEMBER

Number of attributes: 15

Key attributes: Social-Security-Number (SSN)

Foreign keys: Unit-Identification-Code (UIC)

Curriculum-Number

Relationships:	ACTIVITY to MEMBER; 1:N; Mandatory:Optional CURRICULUM to MEMBER; 1:N; Optional:Optional
----------------	---

ACTIVITY

Number of attributes:	4
Key attributes:	UIC
Foreign keys:	None
Relationships:	ACTIVITY to MEMBER; 1:N; Mandatory:Optional

CURRICULUM

Number of attributes:	4
Key attributes:	Curriculum-Number
Foreign keys:	None
Relationships:	CURRICULUM to MEMBER; 1:N; Mandatory:Optional

B. APPLICATION DESIGN

The application is the interface between the user and the database. It contains various control mechanisms to prevent direct access to the database and to maintain the integrity of the database. A menu hierarchy was used to aid and control user interaction with the system. The menu-driven approach was employed because it enables inexperienced end-users to access and use the full functionality of a system faster than with a command-driven system. The menu hierarchy depicted in Figure 4 was derived from user requirements. The Append, Edit/View, and Delete/View sub-menus apply to a selected object database. All user-selectable operations flowed from Main Menu selections. Figure 5 shows the final look of the Main Menu and depicts the generic structure of all menus. Figure 6 provides a view of the form for editing/viewing an existing member record. Although specific fields differ across the various forms in the application, the same form "template" is used throughout the application. Appendix C,

the User's Manual, presents comprehensive graphics of application menus, reports, forms, recall letters, and screens.

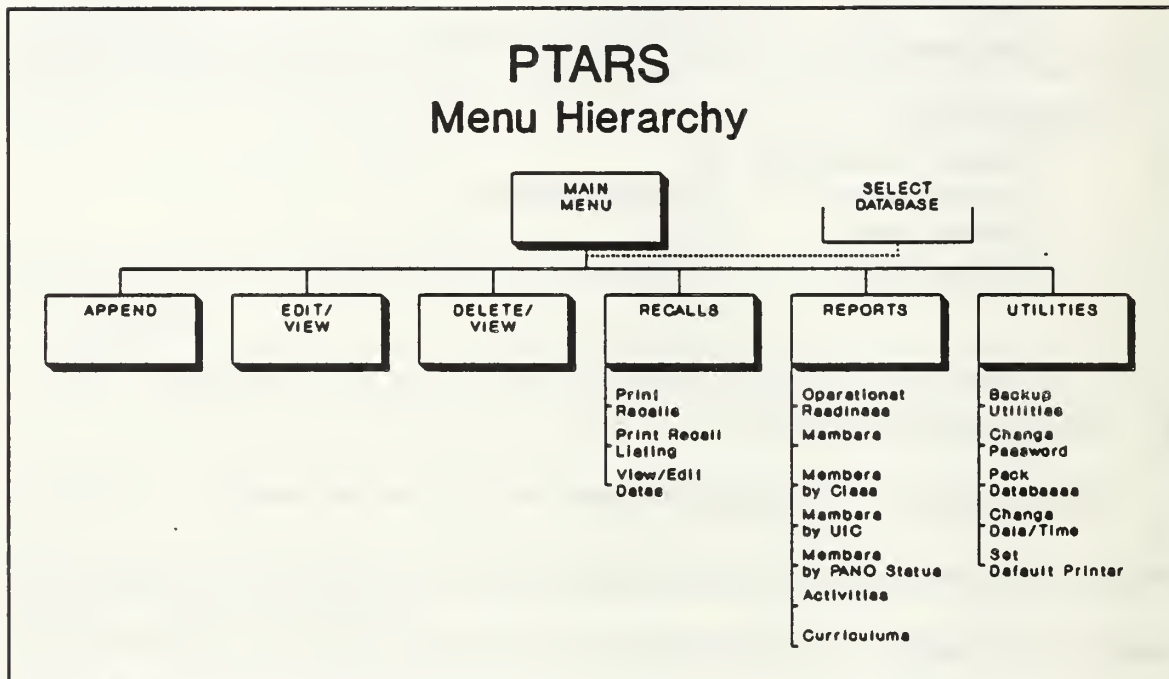


Figure 4. Menu Hierarchy

MEMBER ACTIVITY CURRICULUM DIRECTOR		01/28/92 12:00:00 am
P T A R S M A I N M E N U		
<p><F1> for help <Alt+F1> for functions</p>	<ul style="list-style-type: none"> 0. Quit 1. Append 2. Edit/view 3. Delete/view 4. Recalls ... 5. rePorts ... 6. Select database 7. Utilities ... 	
<div style="border-top: 1px solid black; border-bottom: 1px solid black; display: inline-block;">select : :</div>		

Figure 5. Main Menu Screen

Record: 000013 <MEMBERS >		01/28/92 12:00:00 am												
<p><F1> for Help Member's SSN 123-45-6789</p>														
Last Name Doherty	First Name Janet	M.I. I												
Rank/Rate LT	Service Branch USN	Last T2 Exam 11/21/90 MM/DD/YY												
<p>Pano Status GRN</p>														
UIC 01405	NPS Student Curriculum Number 360	SMC 1000												
<p style="text-align: center;">Dates of Previous Recall Letters Routed To Member</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; width: 25%;">Recall 1</td> <td style="text-align: center; width: 25%;">Recall 2</td> <td style="text-align: center; width: 25%;">Recall 3</td> <td style="text-align: center; width: 25%;">Recall 4</td> </tr> <tr> <td style="text-align: center;">11/21/91</td> <td style="text-align: center;">/ /</td> <td style="text-align: center;">/ /</td> <td style="text-align: center;">/ /</td> </tr> <tr> <td style="text-align: center;">MM/DD/YY</td> <td style="text-align: center;">MM/DD/YY</td> <td style="text-align: center;">MM/DD/YY</td> <td style="text-align: center;">MM/DD/YY</td> </tr> </table>			Recall 1	Recall 2	Recall 3	Recall 4	11/21/91	/ /	/ /	/ /	MM/DD/YY	MM/DD/YY	MM/DD/YY	MM/DD/YY
Recall 1	Recall 2	Recall 3	Recall 4											
11/21/91	/ /	/ /	/ /											
MM/DD/YY	MM/DD/YY	MM/DD/YY	MM/DD/YY											
<p>EDIT/VIEW: <E>dit <F>ind <G>oto <N>ext-record <P>rev-record <Return></p>														

Figure 6. MEMBER Edit/View Form

After establishing the menu hierarchy and obtaining user approval of report, form, recall letter, and screen prototypes, an integrated prototype of the application was developed. That is, a working model of the system was created but with incomplete

functionality [Ref. 3, 4]. End-user evaluations of the prototype's characteristics and operation were used to iteratively revise the model. This prototype was then expanded in functionality to become the final system. This approach was facilitated by Foxpro's project management capability for unifying and coordinating the separate elements of the application. Added advantage was obtained from the use of this approach in that end-users became intimately involved in the development process and actively influenced the look and functioning of the final system. Thus, by the time of implementation their expectations were satisfied and they were well-versed in the application's functioning.

Care was taken to establish consistency of function across modules with regard to form and menu design, messages, escape procedures, navigation keys, function-key use, and availability of on-line help. Moreover, as indicated in the object specifications (Appendix A), the range and format of data for most of the fields was carefully controlled.

IV. SYSTEM IMPLEMENTATION

System implementation was the final step of the development process. The primary objective was to build the fully functional physical application that satisfied the end-user. The physical database was constructed using a DBMS-specific methodology, Foxpro 2.0. It is compatible with the widely-used dBase DBMS language and has numerous language extensions. Moreover, as noted previously, the product provides a very efficient, windowed development environment that facilitates coding, compiling, running, and debugging from within an integrated interface.

During implementation, the prototype was expanded to include all modules fully integrated into an application with complete functionality. Appendix C, the User's Manual, provides documentation which details the final application's features and operations. Documented program code, procedure and token listings, and a token cross-reference listing are included in Appendix E.

Installation required converting the mainframe database and adding several data elements. Hence, the installation and transition to the new system took several days to complete. Primary user training was accomplished during the development process.

V. SUMMARY AND RECOMMENDATIONS

A. SUMMARY

The mainframe-based patient tracking and recall system was due for replacement. It was out-dated in its user interface, was unreliable to access, lacked adequate field validation checks, and required additional capabilities. The PTARS system designed and implemented during the course of this thesis addressed all of these deficiencies and included users actively in the development process. The system is user-friendly and includes all necessary functions internally to provide security, data integrity, and an intuitive operation.

B. RECOMMENDATIONS

During the development process much thought was given to anticipating the needs of end-users. On-line, context-sensitive help was provided for all operations and fields; and confirmations, messages, and prompts were built into all operations that affected the content of the database. Nevertheless, it is still incumbent upon the user to make choices and take actions to protect the data and maintain the quality of unrestricted character fields.

Data security will be only as good as the user's attention to it. The password must be protected, the system must not be left running unattended, and regular backups to floppy disk must be made and stored to safety. All of these activities are ultimately left

up to the discretion of the user. Proper training and careful reading of the User's Manual should enhance end-user adherence to recommended practice.

Finally, NRDC currently provides PC hardware and software support to branch clinics. Upon request, a PC technical expert will troubleshoot problems with BDCM computer resources. The necessity of PCs in the branch clinics is acknowledged and some standard software is provided for an integrated dental information system. Yet, clinics are not provided the resources to protect their systems. For example, no user training is conducted regarding routine machine or data maintenance or security. This could develop into a significant problem in the event of a large data loss. NRDC should consider providing all branch clinics with reasonably efficient backup software, disk maintenance and data recovery software utilities, and the training to use them effectively.

LIST OF REFERENCES

1. *Foxpro Developer's Guide*, Fox Software, Perrysburg, Ohio, 1991.
2. Kroenke, D. M. and Dolan, K.A., *Database Processing* (Third Edition), Science Research Associates, Inc., Chicago, Illinois, 1988.
3. Boar, B., *Application Prototyping: A Requirements Definition Strategy for the 80's*, John Wiley & Sons, New York, New York, 1984.
4. Senn, J. A., *Analysis and Design of Information Systems* (Second Edition), McGraw-Hill, New York, New York, 1989.

APPENDIX A: OBJECT SPECIFICATIONS

Object Definitions

MEMBER OBJECT

<u>Descriptive name</u>	<u>Domain name</u>
Social Security Number	SSN
Last Name	Last_name
First Name	First_name
Middle Initial	MI
Rank or Rate	Rank_rate
Service Branch	Branch
Last T2 Exam	Last_T2
Class Rating	Class
Pano X-ray Status	Pano
SMC or Department Code	SMC
Recall Letter 1 Date	Recall_1
Recall Letter 2 Date	Recall_2
Recall Letter 3 Date	Recall_3
Recall Letter 4 Date	Recall_4
ACTIVITY; ACTIVITY object	
CURRICULUM; CURRICULUM object	

ACTIVITY OBJECT

<u>Descriptive name</u>	<u>Domain name</u>
Unit Identification Code	UIC
Unit Acronym	Acronym
Activity Name	Act_name
Point-of-Contact	POC
MEMBER; MEMBER object; MV	

CURRICULUM OBJECT

<u>Descriptive name</u>	<u>Domain name</u>
Curriculum Number	Curr_num
Curriculum Name	Curr_name
Department Code	Dept_code
Phone Number	Phone_no
MEMBER; MEMBER object; MV	

Domain Definitions

Acronym:

Character (11)
Abbreviated activity name

Act_name:

Character (47)
Official abbreviated name of an NPS tenant command served by BDCM

Branch:

Character (4)
Abbreviation for member's service branch

Class:

Numeric (1), range 1-4
Class rating assigned by dentist to each member

Curr_name:

Character (46)
NPS curriculum name

Curr_num:

Character (3)
Unique NPS curriculum number code

Dept_code:

Character (2)
Curriculum office NPS department code

First_name:

Character (15)
Member's first name

Last_name:

Character (23)
Member's last name

Last_T2:

Date (8); Mask MM/DD/YY, where MM is month, DD is day, YY is year
Last T2 exam date

MI:

Character (1)
Member's middle initial

Pano:

Character (3)
Member's pano x-ray status

Rank_rate:

Character (5)

Member's rank or rate

Recall_1:

Date (8); Mask MM/DD/YY, where MM is month, DD is day, YY is year

Recall letter 1 date

Recall_2:

Date (8); Mask MM/DD/YY, where MM is month, DD is day, YY is year

Recall letter 2 date

Recall_3:

Date (8); Mask MM/DD/YY, where MM is month, DD is day, YY is year

Recall letter 3 date

Recall_4:

Date (8); Mask MM/DD/YY, where MM is month, DD is day, YY is year

Recall letter 4 date

SMC:

Character (4)

Member's student mail center number or staff department mail code

SSN:

Character (11); Mask NNN-NN-NNNN, where N are any digits

Unique member Social Security Number

UIC:

Character (6)

Unique Unit Identification Code of NPS tenant command

APPENDIX B: UPDATE, DISPLAY, AND CONTROL MECHANISMS

I. Update Mechanisms

A. Append/Edit MEMBER data

1. Inputs

- Initial member data received at physical check-in of member records to BDCM
- Member change data received on roster from PSD
- Member change data received on roster from Registrar
- MEMBER object instance from database
- ACTIVITY object instance from database
- CURRICULUM object instance from database
- System-date and time

2. Outputs

- New or modified MEMBER object instance in database
- Confirmation message on screen

3. Processing notes

- This function used for both new and current members
- All initial member data manually entered after review of member's dental record
- Student SMC number may not be available initially

4. Volume

- 225 Jun; 75 Feb/Jul; 250 Mar/Sep/Dec
- Seven per week on average after quarter start
- 275 edits per week on average

5. Frequency

- Six times per year for large batch; otherwise daily

B. Delete MEMBER data

1. Inputs

- Member takes physical custody of dental records upon detachment
- MEMBER objects in database

2. Outputs

- Confirmation notice on screen

3. Processing notes

- Backups of MEMBER data should be made prior to processing a batch of deletions

4. Volume

- 250 at end of each academic quarter
- Seven per week on average after quarter end

5. Frequency

- Four times per year for large batch; otherwise daily

C. Append/Edit ACTIVITY data

1. Inputs

- Activity data change from Personnel Support Detachment (PSD)
- ACTIVITY object instance from database

2. Outputs

- New or modified ACTIVITY object instance in database
- Confirmation message on screen

3. Processing notes

- This function will be seldom used since it will be triggered by the addition or modification of a generally stable client organization
- This function used for both new and current activities
- 4. Volume
 - Variable, approximately one instance every two years on the average
- 5. Frequency
 - Variable, approximately once every two years
- D. Delete ACTIVITY data
 1. Inputs
 - Activity data change from Personnel Support Detachment (PSD)
 - ACTIVITY object instance from database
 2. Outputs
 - Confirmation notice on screen
 3. Processing notes
 - This function will be seldom used since it will be triggered by the elimination of a generally stable client organization
 - Backup of ACTIVITY data should be made prior to deletion
 4. Volume
 - Variable, approximately one instance every four years on the average
 5. Frequency
 - Variable, approximately once every four years
- E. Append/Edit CURRICULUM data
 1. Inputs
 - Curriculum data change from Registrar
 - CURRICULUM object instance from database
 2. Outputs
 - New or modified CURRICULUM object instance
 - Confirmation message on screen
 3. Processing notes
 - This function will be seldom used since it will be triggered by the addition or modification of generally stable curriculums
 - This function used for both new and current curriculums
 4. Volume
 - Variable, approximately two instances per year on the average
 5. Frequency
 - Variable, approximately twice per year prior to new student class
- F. Delete CURRICULUM data
 1. Inputs
 - Curriculum data change from Registrar
 - CURRICULUM object instance from database
 2. Outputs
 - Confirmation message on screen
 3. Processing notes
 - This function will be seldom used since it will be triggered by the elimination of a generally stable curriculum
 - Backup of curriculum data should be made prior to deletion
 4. Volume
 - Variable, approximately one instance every five years on the average
 5. Frequency
 - Variable, approximately once every five years

II. Display Mechanisms

A. Query on MEMBER

1. Output description
 - Form showing all data for a member to screen
2. Source data
 - MEMBER object
 - Member SSN or name keyed by user
3. Processing notes
 - Used by Administrative Petty Officer or Receptionist
4. Volume
 - Five per week
5. Frequency
 - Daily

B. Recall letter 1

1. Output description
 - Memorandum mailed to member
 - New or modified MEMBER object instance in database
2. Source data
 - MEMBER object
 - System-date
3. Processing notes
 - This process is initiated from a menu by the user. It creates recall letter one for all members whose last T2 exam was more than 10 months prior to the system-date and for whom recall letter one was not previously produced
 - This process inserts system-date as Recall-Ltr1-Date when conditions above exist
4. Volume
 - 160 monthly
5. Frequency
 - End of every month

C. Recall letter 2

1. Output description
 - Memorandum mailed to member
 - New or modified MEMBER object instance in database
2. Source data
 - MEMBER object
 - System-date
3. Processing notes
 - This process is initiated from a menu by the user. It creates recall letter two for all members whose last T2 exam was more than 11 months prior to the system-date, for whom recall letter one was produced, and for whom recall letter two was not previously produced
 - This process inserts system-date as Recall-Ltr2-Date when conditions above exist
4. Volume
 - 100 monthly
5. Frequency
 - End of every month

D. Recall letter 3

1. Output description
 - Letter mailed to member
 - New or modified MEMBER object instance in database

2. Source data
 - MEMBER object
 - System-date
 3. Processing notes
 - This process is initiated from a menu by the user. It produces recall letter three for all members whose last T2 exam was more than 12 months prior to the system-date, for whom recall letter two was produced, and for whom recall letter three was not previously produced
 - This process inserts system-date as Recall-Ltr2-Date when conditions above exist
 4. Volume
 - 30 monthly
 5. Frequency
 - End of every month
- E. Recall letter 4
1. Output description
 - Letter mailed to Curriculum Officer for student members and Activity POC for all other members
 - New or modified MEMBER object instance in database
 2. Source data
 - MEMBER object
 - ACTIVITY object
 - CURRICULUM object
 - System-date
 3. Processing notes
 - This process is initiated from a menu by the user. It produces recall letter four for all members whose last T2 exam was more than 13 months prior to the system-date, for whom recall letter three was produced, and for whom recall letter four was not previously produced
 - This process inserts system-date as Recall-Ltr4-Date when conditions above exist
 - Student members uniquely belong to UIC 31405
 4. Volume
 - 3 monthly
 5. Frequency
 - End of every month
- F. Operational Readiness Report
1. Output description
 - Screen display of summary count and percent of patient load for all members by class
 - Screen display of summary count and percent of all patients in Pano x-ray status categories
 2. Source data
 - MEMBER object
 - System-date
 3. Processing notes
 - This process is initiated from a menu by the user. It creates a summary report of the number and percentage of all members in each of the four different dental classes. The report can be optionally printed.
 4. Volume
 - 1 monthly
 5. Frequency
 - End of every month

G. Member Roster

1. Output description
 - Printed roster of all members sorted alphabetically or by SSN
2. Source data
 - MEMBER object
 - System-date
3. Processing notes
 - This process is initiated from a menu by the user.
4. Volume
 - 1 monthly
5. Frequency
 - End of every month

H. Member Roster by Class

1. Output description
 - Printed roster of members sorted alphabetically or by SSN; available for all or for specified class
2. Source data
 - MEMBER object
 - System-date
3. Processing notes
 - This process is initiated from a menu by the user.
4. Volume
 - 1 monthly
5. Frequency
 - End of every month

I. Member Roster by UIC

1. Output description
 - Printed roster of all members sorted alphabetically or by SSN
2. Source data
 - MEMBER object
 - System-date
3. Processing notes
 - This process is initiated from a menu by the user.
4. Volume
 - 1 monthly
5. Frequency
 - End of every month

J. Member Roster by Pano X-ray status

1. Output description
 - Printed roster of members sorted alphabetically or by SSN; available for all members or for specified Pano status
2. Source data
 - MEMBER object
 - System-date
3. Processing notes
 - This process is initiated from a menu by the user.
4. Volume
 - 1 monthly
5. Frequency

- End of every month

K. Activities Listing

1. Output description
 - Printed roster of Activities sorted by UIC
2. Source data
 - ACTIVITY object
 - System-date
3. Processing notes
 - This process is initiated from a menu by the user.
4. Volume
 - 1 monthly
5. Frequency
 - End of every month

L. Curriculum Listing

1. Output description
 - Printed roster of Curriculum Listing sorted by curriculum number
2. Source data
 - CURRICULUM object
 - System-date
3. Processing notes
 - This process is initiated from a menu by the user.
4. Volume
 - 1 monthly
5. Frequency
 - End of every month

III. Control Mechanisms

- A. Access to the system is protected by a password known only by the Administrative Petty Officer and the Receptionist
- B. The system is limited to use by one person at a time.
- C. Monthly validations of various member data are accomplished using rosters obtained from PSD and the Registrar

APPENDIX C: USER'S MANUAL

NPS DENTAL CLINIC PATIENT TRACKING & RECALL SYSTEM



Timothy P. Steele
LCDR, MSC, USN
Naval Postgraduate School

CONTENTS

<i>Introduction</i>	<i>1</i>
Features overview	1
Typographical conventions	2
 <i>Chapter 1 Getting started</i>	 <i>3</i>
System requirements	3
Installation	4
Installation overview	4
Installing PTARS	4
Re-installing PTARS	5
Starting PTARS	6
Updating member CLASS	6
Security	6
Creating a start-up batch file	7
 <i>Chapter 2 Getting around</i>	 <i>9</i>
Navigation/Activation keys	9
Function keys	9
Using on-line Help	10
Menus overview	11
Main Menu	12
Selecting a database	12
Exiting PTARS	13
 <i>Chapter 3 Database updating</i>	 <i>15</i>
Appending records	15
Editing/viewing records	16
Deleting/viewing records	19
 <i>Chapter 4 Recalls</i>	 <i>21</i>
Printing recalls	22
Printing recalls lists	27
Viewing/editing recall dates	27

<i>Chapter 5 Reports</i>	29
Operational readiness	29
Rosters	30
<i>Chapter 6 Utilities</i>	33
Backup utilities	33
Backing up database(s)	34
Listing files	35
Formatting a floppy disk	35
Restoring backup(s)	35
Changing the password	36
Packing the database(s)	37
Changing the date or time	37
Selecting the default printer	37
<i>Appendix A Optimizing PTARS</i>	39
Disk defrag/compress	39
Memory	39
Conventional memory	39
Expanded memory	40
Extended memory	40
Config.sys	41
Pack the database(s)	41
<i>Appendix B File definitions</i>	43
<i>Appendix C Database specifications</i>	45

Introduction

Welcome to the Naval Postgraduate School Dental Clinic (NPSDC) Patient Tracking and Recall System (PTARS). This database application was developed to provide an in-house, PC-based capability for NPSDC to maintain the patient data necessary to track and recall patients for annual exams and to produce operational readiness statistics. The system provides fast, dependable access to member records and automates the recall process.

PTARS was designed based on extensive interviews with the NPSDC staff to identify clinic requirements. Prototypes of the system were iteratively developed and demonstrated to ensure that clinic end-users were fully satisfied with the final system specifications. A primary design objective was to develop an application that was very user-friendly. Hence, you will be able to use the system productively with only a minimum amount of familiarization time. Please take a few minutes now to review this User's Manual.

Features overview

PTARS employs four database files that are directly accessible to user modification: MEMBERS.DBF, ACTIVITY.DBF, CURRICUL.DBF, and DIRECTOR.DBF. MEMBERS.DBF contains the information pertinent to each patient. The files ACTIVITY.DBF and CURRICUL.DBF are used for locating patients and for printing recall letter addresses. ACTIVITY.DBF contains information specific to each UIC served by NPSDC and CURRICUL.DBF contains information specific to each NPS student curriculum. DIRECTOR.DBF contains the name of the current NPSDC Director for placement into the signature line of recall letters.

The application provides a series of simple menus and sub-menus from which to choose its various options. You will be able to view, append, update, and delete Member, Activity, Curriculum, and Director data using screen forms with built-in error-checking routines for each action or data entry. You will also be able to print special reports, sorted database listings, and recall letters. Additional features include but are not limited to:

- Password controlled access to PTARS; changeable password
- Automatic updating of member treatment class status

- Context-sensitive help
- System information display
- Continuous date and time display
- Automatic determination of appropriate recall letters to print
- Backup database(s) to hard disk or floppy disk; restore backup(s)
- Format floppy disk from within application
- List files on hard disk or floppy disk
- Automatic reminders for database backup (if more than one month since last backup) and database pack (if more than 10% of records marked for deletion)

Typographical conventions

The following typographical conventions are used in this manual:

Input Anything that you type is in the Courier typeface, for example,
a:\setup <Enter>

Keys Keys to be pressed are represented like this:
<Esc> <Enter> <F1> {C}

Press both keys simultaneously when a "+" symbol is present, as in:
<Alt+F1>

Direction Cursor movement keys are indicated as:
<PgUp> <PgDn> <Arrows>



Getting started

This chapter contains all the information you need to install and run PTARS. It also discusses the various settings that you can change.

It contains the following sections:

- System requirements
- Installation
- Starting PTARS
- Creating a batch file

System requirements

PTARS requires the following hardware and software:

- An IBM compatible computer with at least 512K of random access memory (RAM) (640K of RAM strongly recommended)
- One floppy disk and one hard disk drive (with at least 3 megabytes of space available)
- Version 2.0 or later of DOS
- A CONFIG.SYS file in your root directory with a Files=25 (or greater) statement
- An EGA or VGA video adapter
- An Epson E/F/J/RX/LQ compatible or IBM Proprinter compatible dot-matrix printer

Additional requirements:

- To take advantage of Expanded memory support, you need an expanded memory card that is hardware and software compatible with the Lotus-Intel-Microsoft standard 4.0 or later (LIM 4.0 EMS). If you have an Intel 80386 or 80486 processor you can also use extended memory and a software expanded memory emulator program. PTARS can use 64K

of expanded memory as additional general purpose memory and any remaining expanded memory to speed up file I/O.

- If expanded memory is not available but the computer has extended memory, PTARS can be configured during installation to use 512K of the available extended memory for a disk cache to speed up file I/O.
- Double-copy paper to automatically make copies of recall letters. Since a copy of Recall 3 is identified as an enclosure to Recall 4, a copy of Recall 3 should be available before routing Recall 4. An alternative to double-copy paper would be making a copy of all Recall 3 letters after printing; then filing them in the event a Recall 4 was necessary for the same individual(s).

Installation

Installation overview

You have been provided with four numbered floppy disks. Disks 1 to 3 contain the files necessary to install and run PTARS. Disk 4 contains the initial database files that were current at the time of program delivery (i.e., MEMBERS.DBF, ACTIVITY.DBF, CURRICUL.DBF, and DIRECTOR.DBF). There are two steps to installing PTARS:

- **Make a backup and install the program.** Before you do anything else, copy the original disks and store them in a safe place. Then, use your copies of the original disks and run the Setup program to install PTARS on your hard disk.
- **Choose the default printer.** Before you print for the first time, you should select the default printer emulation from the Utilities Menu.

Installing PTARS

Refer to your computer's documentation (or ask your local computer guru) to determine whether your computer has expanded memory, disk caching hardware or software, and/or extended memory. You will be queried during the installation process regarding your computer's configuration. Note that you need at least 3 megabytes of available hard disk space before you begin.

One cautionary note before beginning your installation. PTARS was designed to run using only one computer at a time. Although in the future it may be tempting to install PTARS on a second computer, **avoid installing PTARS on more than one computer.** Because the separate installations can not communicate, there is no built-in, guaranteed way for the separate databases to maintain the same up-to-date data. Although you could

theoretically transfer data using floppy disks, almost assuredly over time some data would exist in one machine but not the other, and vice-versa.

The steps for installing PTARS are as follows:

1. Insert the PTARS disk #1 in drive A.
2. At the DOS prompt, type `a:\setup`. The Setup program will start.
3. When prompted by Setup, specify the disk where you want to install PTARS (e.g., c). Setup creates the subdirectory "`\PTARS`" on the hard disk specified and copies the program files and initial database files to it. Setup prompts you to insert each disk when necessary.
4. After copying, assembling, and un-compressing all the files from the installation disks, Setup queries whether your computer has expanded memory and/or a disk cache. Respond `y` or `n`, as appropriate. If you respond negatively, Setup queries whether you have extended memory. Again, respond as appropriate. This process determines how PTARS is configured for start-up.
5. When the installation is complete, Setup presents a screen with installation notes. Read the notes. Setup then queries whether you want to start PTARS. If you respond affirmatively, PTARS loads immediately.
6. Before printing from PTARS for the first time, select the default printer from the Utilities Menu. Refer to your printer's documentation to determine which emulation (Epson E/F/J/RX/LQ or IBM Proprinter) your printer uses. The default printer emulation is initially Epson.
7. Align the paper in your printer. Test the margin adjustments of your paper by printing the Operational Readiness Report from the Reports Menu. The top of your paper should be set in your printer so that one blank line exists at the top of the printed report. Likewise, the paper should be set so that one blank space exists to the left of the header statement "FOR OFFICIAL USE ONLY". If your paper is adjusted in the printer to satisfy these conditions, all printing from PTARS will be formatted properly.

Re-installing PTARS

There are two instances when you may want to re-install PTARS: 1) when there is some problem with any of the program files or 2) the computer has been modified with regard to expanded memory, a disk cache, or extended memory.

The re-install process is exactly the same as the initial installation with two exceptions. Setup attempts to determine if PTARS has been installed previously. If Setup detects that this is a re-installation you will be presented with a listing of existing database files in the "\PTARS" subdirectory and a re-installation note on screen. You can elect to continue or quit the re-installation at this time. If you elect to continue, you will be queried regarding which, if any, of the initial database files you may want to re-install. Note that if you have been using PTARS for any period of time you will probably elect not to re-install any of the initial database files. This is because they will be out of date. Use the "Restore backup(s)" option in the Backup Utilities Menu to restore your most recent data from floppy disk, if necessary.

Starting PTARS

If necessary, change to the "\PTARS" subdirectory on the drive where you installed PTARS (e.g., at the DOS prompt, type `cd\ptars`). Then type `ptars` and press <Enter>. A logo screen will appear and pause briefly. (You can eliminate the pause by pressing any key during the logo display.) Following the pause, the PTARS Access Screen appears and you are requested to enter the password. The initial password to use is "zyxabc". You will be given up to three attempts to enter the correct password. After a third failure, PTARS shuts down.

After correctly entering the password, you will be queried whether the system date and time are correct. If you respond negatively, you are prompted to enter the correct date and/or time according to the format displayed.

Updating member CLASS

When the system date and time are correct, PTARS updates each member's dental CLASS rating. CLASS ratings of "1", "2", or "3" are assigned to members by an examining dentist. A CLASS rating of "4" indicates simply that the member is due for his/her mandatory annual dental examination. PTARS scans each record in the MEMBERS.DBF database file and checks to see if the LAST_T2 date is more than 12 months prior to the current system date. If so, it replaces the existing CLASS rating with "4". After updating member CLASS, PTARS displays the Main Menu.

Security

It is *strongly* recommended that the default password be changed after installing the PTARS program. Your data is extremely important. Inadvertent or deliberate tampering with your data by an unauthorized person can only be prevented by taking security precautions (*and* taking them seriously). In addition to keeping a secure password, it is very important that you do not leave PTARS running unattended. The temptation to do

so, however, will be great. Making regular backups of your data to floppy disk and putting them in a safe place is probably the best way to ensure against loss of data due to any cause.

Creating a start-up batch file

A DOS batch file can be created that will enable you to start PTARS at any time regardless of what directory you may currently be in, without having to type additional DOS commands. Use a text editor (or a word processor mode that does not insert hidden formatting codes) to create a batch file like the example below. The example batch file assumes that you have installed PTARS to the C drive.

```
C:
CD\PTARS
PTARS
```

When the batch file is complete, name it "PTARS.BAT" and place it in your root directory or any directory that is in your DOS path. Henceforth, simply type PTARS to load the PTARS program from any location. See any DOS reference for terminology assistance.



Getting around

This chapter contains the information you need to navigate the menus, forms, and fields of PTARS. It covers:

- Navigation/Action keys
- Function keys
- Using on-line Help
- Menu overview
- Main menu
- Exiting PTARS

Navigation/Action keys

Each PTARS screen shows the available commands or options. The following keys let you move around a screen, between or within fields, or perform various generic actions:

<u>Press:</u>	<u>To:</u>
<Arrows>	move up or down one line; move left or right one character or screen
<PgUp>/<PgDn>	display previous or next screen of a multiple records screen
<Home>	move to the start of a multiple records screen or input field
<End>	move to the end of a multiple records screen or input field
<Backspace>	delete character to left; move back one input field
<Return>	accept an entry; move to next field
<Insert>	toggle insert/typeover mode
	delete a character or record
<Esc>	cancel the current task

Function keys

Function keys <F1> through <F4> are assigned specific actions as described below. Pressing <Alt+F1> (pressing both keys simultaneously) at any time presents a popup reminder list of the functions available. Functions are activated by pressing the assigned

function key or selecting the function from the popup list. Functions are available at all times, regardless of the current activity. The functions available are:

- Help <F1>** Context-sensitive help window. See the next section, "Using on-line Help".
- Calendar <F2>** Pops-up a monthly Calendar display. It shows the current month in row and column form with the current day highlighted. You can move forward or backward in months by pressing <PgUp> or <PgDn>, and in years by pressing <Ctrl-PgUp> or <Ctrl-PgDn>, respectively. To get back to the current date, press {T}. As with almost all operations in PTARS, press <Esc> to exit.
- Poptris <F3>** A Tetris-like diversion. The object is to fill the rectangular field with the falling objects from the bottom up without leaving any open spaces. Use the numeric keypad arrows to position falling objects within the field. Pressing the number 5 key causes the shape of the falling object to change. It can be pressed repeatedly to cycle the shape of the falling object. Pressing the ↓ arrow key causes the falling object to land immediately, hence, speeding up the activity. Additional commands/functions are displayed on-screen. Poptris code has been included by permission of Gerald F. Garcia.
- About PTARS <F4>** A window containing system environment information. It includes information on the operating system, computer hardware, RAM, and disk space.

Using on-line Help

On-line Help is available at all times by pressing <F1>. Help is "context-sensitive" since the Help Topic details initially displayed apply to the current PTARS screen. When the † symbol is present in the topic box, you can scroll down or up through the Help window to view additional text using the ↓ or ↑ arrow keys.

As shown in Figure 1, the Help window consists of two panels—one lists Help Topics and the other displays details about each Topic. At the bottom of the Topics list all fields in the various databases are identified with a " ~ " prefix and are defined. Commands available in Help are described below:

- « Topics » This provides a list of Topics available in the Help system. To select a Topic you can: 1) use the arrow keys to scroll through the Topics

to find the one you want or 2) type a letter or series of letters to select the first Topic beginning with those letter(s). To see details about a Topic, select the Topic and press <Return>.

- <Next> This selects Help details for the next Topic in the help file list.
- <Previous> This selects Help details for the prior Topic in the help file list.
- <Look Up> Enables you to find the closest Topic match to a word that you highlight within Help details. When you highlight a word in the Help text, the <Look up> function becomes available. You highlight a word by placing the cursor at the first letter in a word using the ← and → arrow keys. Then press <Shift+→> to highlight the word.
- See Also This lists Help Topics that may be of interest related to the current Topic.
- <Esc> Exits Help.

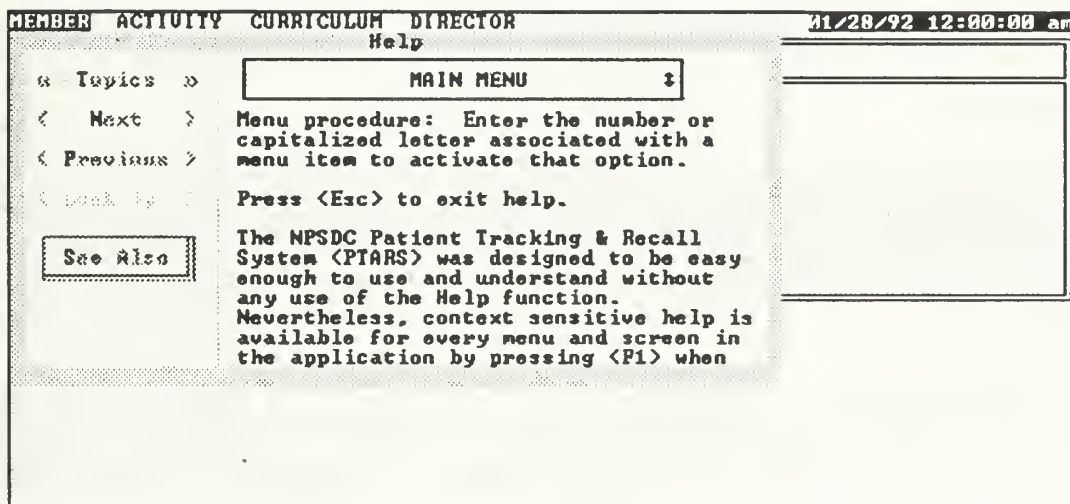


Figure 1. Help window appearing over Main Menu.

Menus overview

PTARS is a "menu-driven" system. All operations are activated by selecting options from full-screen menus, from sub-menus located at the bottom of the screen, or from pop-up menus. An option can be selected on all menus by pressing the highlighted (and capitalized) letter associated with the option. On full-screen menus the number of the menu option will also activate the option. On popup menus you can also scroll to the

desired option and press <Enter> to activate the option. Figure 2 below provides a graphical view of the major menu operations within PTARS.

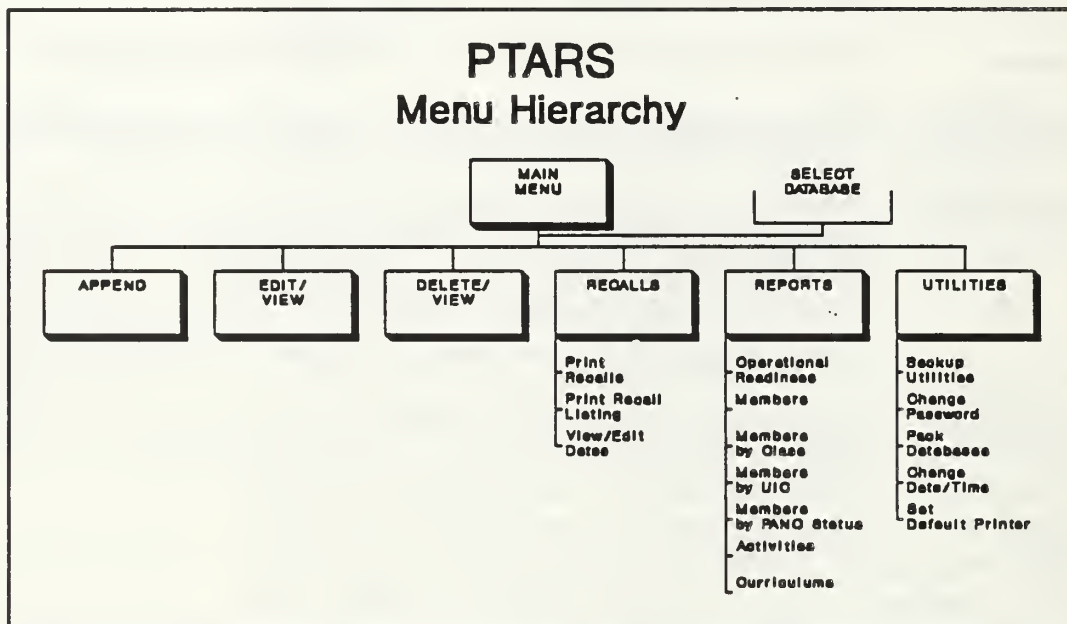


Figure 2. PTARS menu hierarchy.

Main Menu

After updating member CLASS, PTARS displays the Main Menu, as shown in Figure 3 on the next page. Each screen in PTARS continuously displays the system date and time in the upper right corner.

Selecting a database

In the upper left corner of the Main Menu the four databases of interest are identified. The active database is highlighted and blinking. By default, Members is the initially active database. The Main Menu options "Append", "Edit/view", and "Delete/view" apply only to the active database. A different database can be made active by choosing the option, "Select database", and then selecting the desired database from the popup selection list.

Exiting PTARS is discussed in the following sub-section. The remaining Main Menu options are covered in detail in subsequent chapters.

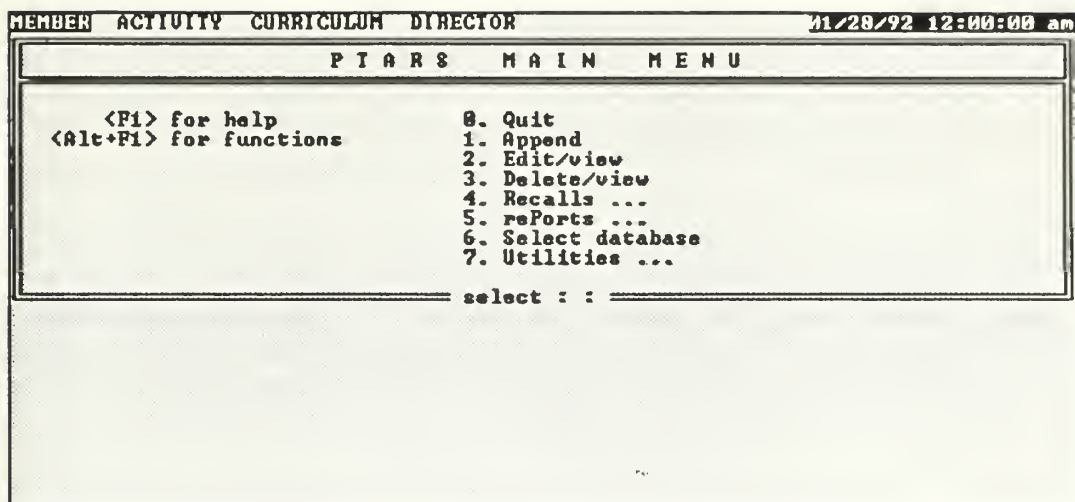


Figure 3. PTARS Main Menu.

Exiting PTARS

It is very important that you exit (quit) PTARS using the Main Menu "Quit" option. If you reboot the computer with <Ctrl+Alt+Del> or shut the power off without first quitting properly, any databases which are in use at the time are vulnerable to damage. Hence, it is essential that you exit only by using the Main Menu "Quit" option.

When quitting, several things happen before the system shuts down. First, PTARS checks to see if it has been more than one month since MEMBERS.DBF has been backed-up to a floppy disk. If so, a reminder message pops-up on screen and you are given the option to perform a backup. If you choose to perform a backup, PTARS switches to the Backup Utilities Menu where you can perform your backup operations and quit when you are finished.

Next, PTARS checks to see if more than 10% of the records in MEMBERS.DBF have been marked for deletion. If so, a message pops-up and you are queried whether you want to "pack" the database. See Chapter 6 for details on packing the database.

Finally, before shutting down, PTARS queries whether you want to back-up the databases to the hard disk. This allows you to save a second copy of your session's work on the hard disk. See Chapter 6 for further coverage of backing-up.



Database updating

This chapter contains the information necessary for updating the databases by appending, editing, or deleting records. Several example screens will be shown to preview the look of PTARS when working with its various modes.

Appending Records

Select the "Append" option from the Main Menu to append records. Appending records involves adding new records to a database. New records can be appended to MEMBERS.DBF, ACTIVITY.DBF, and CURRICUL.DBF. Unlike the foregoing three databases, DIRECTOR.DBF contains only one record. This record contains the name of the current clinic director and must always be present. Hence, it can only be edited.

As discussed in Chapter 2, PTARS starts by default with MEMBERS.DBF as the active database. You can select a different database from the Main Menu option "Select Database". To append records, press {A} from the Main Menu. A blank form will appear, ready to receive new data. You can abort from appending by pressing <Esc> and the record will not be saved.

When appending a record almost all fields require an entry. If a field is left blank and <Enter> is pressed, either a warning will appear stating that an entry is required or a popup list of valid field entries will appear. When a popup list appears, scroll to the desired field entry and press <Enter> to insert the entry into the form. Figure 4 shows the Append data entry form for Members.

If the member is an NPS student (i.e., UIC = "31405"), a field for Curriculum Number and SMC (Student Mail Center number) will appear following UIC. Alternatively, if the member is a non-student, a field for Activity Department Code will appear. Enter data into these fields as appropriate.

As a reminder, if you have any doubts regarding the contents of a certain field, be sure to utilize the Help function. Each field in all the databases is described in the Topics section of Help. Field names are prefixed with the "~" symbol and are located at the bottom of the scrollable Help Topics list.

Record: 001801		<MEMBERS >		*BLANK*		01/28/92 12:00:00 am	
<F1> for Help Member's SSN <input type="text"/>							
Last Name <input type="text"/>		First Name <input type="text"/>		M.I. <input type="text"/>			
Rank/Rate <input type="text"/>		Service Branch <input type="text"/>		Last T2 Exam <input type="text"/>		Class <input type="text"/>	
Pano Status <input type="text"/>				MM/DD/YY			
UIC <input type="text"/>							
APPEND: Press <Esc> to abort							

Figure 4. Append record form for Members in append mode.

After completing the data entry for a new record or after aborting an append, a sub-menu will appear at the bottom of the screen with several options:

<Return>:add-another {E}dit {F}inished

Pressing <Return> brings up a blank form for appending another new record. Pressing {E}dit allows editing of the currently displayed record. Selecting {F}inished appends the record (if completely entered and not marked for deletion) and returns you to the Main Menu. Pressing toggles between deleting and saving the current record. For example, assume you discover an error in a record that you have just entered and you want to delete it so that you can get the correct info later and re-enter it. Press to delete it. This allows you to then press <Enter> to keep entering new records without saving the erroneous one. When a record is "Deleted" a status indicator at the top of the screen says " *Deleted* ". In the next section, forms for editing each of the databases will be displayed. The forms look very similar to the forms for appending data.

Editing/viewing records

The "Edit/view" option of the Main Menu allows you to edit records in the active database. Editing is performed with one record displayed at a time. This option also provides a means to view all the data in a record of the active database on a single screen.

As can be seen in Figure 5, the Edit/view form for Members is very similar to the Append form for Members. The difference is that the sub-menu of options available is more extensive and that additional information is shown on the form. In the lower

portion of the Edit Members form the dates of recall letters previously printed to the Member are displayed. This information can not be edited from the Edit/view screen but is for viewing only. Editing of recall dates will be discussed in Chapter 4.

Record: 000013		<MEMBERS>		11/28/92 12:00:00 am																	
<F1> for Help																					
Member's SSN 123-45-6789																					
Last Name Doherty		First Name Janet		M.I. 1																	
Rank/Rate 11	Service Branch 15N	Last T2 Exam 11/21/90		Class 3																	
Pano Status GRN		MM/DD/YY																			
UIC 01405		NPS Student Curriculum Number 0160		SMC 1000																	
<table border="1"> <tr> <th colspan="4">Dates of Previous Recall Letters Routed To Member</th> </tr> <tr> <td>Recall 1</td> <td>Recall 2</td> <td>Recall 3</td> <td>Recall 4</td> </tr> <tr> <td>11/21/91</td> <td>MM/DD/YY</td> <td>MM/DD/YY</td> <td>MM/DD/YY</td> </tr> <tr> <td>MM/DD/YY</td> <td></td> <td></td> <td></td> </tr> </table>						Dates of Previous Recall Letters Routed To Member				Recall 1	Recall 2	Recall 3	Recall 4	11/21/91	MM/DD/YY	MM/DD/YY	MM/DD/YY	MM/DD/YY			
Dates of Previous Recall Letters Routed To Member																					
Recall 1	Recall 2	Recall 3	Recall 4																		
11/21/91	MM/DD/YY	MM/DD/YY	MM/DD/YY																		
MM/DD/YY																					
EDIT/VIEW: <E>dit <F>ind <G>oto <N>ext-record <P>rev-record <Return>																					

Figure 5. Edit/view form for Members.

The actions of each of the Edit/view sub-menu commands are as follows:

- {E}dit** {E}dit returns the cursor to the record displayed for further changes; the sub-menu options are not available. Entry of data in edit mode is the same as when appending a new record. Pressing <Esc> when in edit mode aborts the edit and the original data is displayed.
- {F}ind** When editing Members, {F}ind enables you to select a specific record by specifying a member's SSN or name. (Part of a name or even a single letter can be used. PTARS will seek the first instance of whatever you type. Specifying the person's full name provides an exact match.) Since a name is not necessarily unique, the first occurrence of a match is shown on the screen. Specify a UIC when editing an Activity and a Curriculum Number when editing a Curriculum.
- {G}oto** {G}oto enables you to go to a specific record number in the database. Record numbers are listed in the top left of the edit screen.
- {N}ext** {N}ext-record brings up the next record. (By default, records are sorted by SSN. When a record is "found" by name, the database is sorted by last-name + first-name.)

{P}rev {P}rev-record brings up the prior record. Records are sorted as noted above.

<Return> <Return> brings you back to the Main Menu.

Figures 6 and 7 display the Edit/view forms for the Activity and Curriculum databases, respectively. The Append forms for these databases look the same with the exception of the sub-menus.

Record: 000002		<ACTIVITY>		11/28/92 12:00:00 am									
<div><F1> for Help</div> <table><tr><td>UIC</td><td>Activity Name</td></tr><tr><td>01405</td><td>NPS MONTEREY STUDENT</td></tr><tr><td>Acronym</td><td>Point of Contact</td></tr><tr><td>NPS_STUDENT</td><td>Curriculum Officer</td></tr></table>						UIC	Activity Name	01405	NPS MONTEREY STUDENT	Acronym	Point of Contact	NPS_STUDENT	Curriculum Officer
UIC	Activity Name												
01405	NPS MONTEREY STUDENT												
Acronym	Point of Contact												
NPS_STUDENT	Curriculum Officer												
EDIT/VIEW: <E>dit <F>ind <G>oto <N>ext-record <P>rev-record <Return>													

Figure 6. Edit/view form for Activity.

Record: 000001		<CURRICUL>		11/28/92 12:00:00 am									
<div><F1> for Help</div> <table><tr><td>Curriculum #</td><td>Curriculum Name</td></tr><tr><td>060</td><td>Operations Analysis</td></tr><tr><td>Department</td><td>Phone #</td></tr><tr><td>Code 00</td><td>2786</td></tr></table>						Curriculum #	Curriculum Name	060	Operations Analysis	Department	Phone #	Code 00	2786
Curriculum #	Curriculum Name												
060	Operations Analysis												
Department	Phone #												
Code 00	2786												
EDIT/VIEW: <E>dit <F>ind <G>oto <N>ext-record <P>rev-record <Return>													

Figure 7. Edit/view form for Curriculum.

Figure 8 shows the Edit/view form for Director. As discussed, Director can not be appended to or deleted. Hence, you are automatically in edit mode when you select this form. This is because there is only one clinic Director record and it must always contain a signature name.

The screenshot shows a window titled "<DIRECTOR>" with a timestamp "11/28/92 12:00:00 am" in the top right corner. Inside the window, there is a header bar containing "<F1> for Help" on the left and "Enter new Director: R. C. TERNUNE" in the center. Below this header is a large, empty rectangular area for editing. At the bottom of the window, a status bar displays "EDIT: Press <Esc> to abort".

Figure 8. Edit/view form for Director.

Deleting/viewing records

Select the "Delete/view" option from the Main Menu to delete record(s) or to view multiple records on one screen. When a record is marked for deletion, an "*" appears to the left of the record. Figure 9 shows the Delete/view screen for Members. The Delete/view screens for Activities and for Curriculums operate in the same fashion as for Members. The only difference is the fields displayed on screen. When the "=>" appears in the upper right of the screen on the field column header line, additional fields exist for viewing. Pressing the right arrow key will pan the screen right to view the additional fields. Press the left arrow key to pan back to the left.

When a record is "Deleted" on the Delete/view screen, the record is not actually physically removed from the database; it is simply "marked" for deletion. This means that the record can still be recovered if you decide later that you want to "undelete" it. See the discussion of the action below for its operation. To permanently (physically) remove record(s) from a database, the database must be "packed". Chapter 6, "Utilities", provides further discussion of packing the database.

File: MEMBERS.DBF			11/28/92 12:00:00 am			
<F1> for help			DELETE/VIEW RECORDS			
Record#	SSN	LAST NAME	FIRST NAME	MI	RANK/RATE	-->
1	000-00-0002	Merman	Ethel		LT	
2	001-00-0003	Miserables	Les		LT	
3	012-12-1212	Andrews	Antoine	R	LT	
4	012-93-8475	Adams	John	Q	ENS	
5	022-20-0000	Marcos	Imelda		LTJG	
6	023-12-3122	Wine	Dandelion		ENS	
7	039-39-2828	Lincoln	Mark		ENS	
8	076-35-3746	Bloch	Robert	O	LCDR	
9	083-82-7827	Mathews	Mark	M	LTJG	
10	089-64-3585	Morrison	Larry	R	LTJG	
11	102-20-0000	Mastroiani	Marcello	O	LT	
12	109-28-3746	Laverne	Shirley		DT2	
13	123-45-6789	Doherty	Janet	I	LT	
14	123-58-9213	Madison	James	F	CAPT	
15	123-92-9292	Alexander	Hamilton	A	ENS	
16	133-21-3838	Zawfir	Jonathan	L	SGT	
17	134-15-6789	Sullivan	Karen	I	LTJG	
18	138-38-3838	Mears	Rick		LT	
DELETE/VIEW: {F}ind {G}oto {M}ode <Arrows> <PgDn> <PgUp> <Return>						

Figure 9. Delete/view screen for Members.

The actions of each of the Delete/view sub-menu commands are as follows:

- {F}ind Performs the same action as with the Edit/view form.
- {G}oto Performs the same action as with the Edit/view form.
- {M}ode {M}ode pops-up a selection of display modes for EGA and VGA video adapters: EGA, 25 or 43 lines; VGA, 25 or 50 lines. More lines on a screen are useful when deleting many members in a single session.
- <Arrows> <Arrows> refers to the direction keys for moving sideways to view panels of fields or up and down to place the cursor on different records.
- <PgDn> <PgDn> takes you to the next screen of consecutive records.
- <PgUp> <PgUp> takes you to the prior screen of consecutive records.
- toggles a deletion marker for a record. To mark a record for deletion, move the cursor to the record and press . When a record is marked for deletion an "*" appears to the left of the record. To unmark a deletion, make sure the cursor is on the correct marked record and press again.
- <Return> <Return> brings you back to the Main Menu.

Recalls

Recalls are the primary reason for the existence of PTARS. Each of the Service Branches require that members receive an annual dental examination (a "T2" exam), regardless of any prior need for dental treatment. Hence, members require notification prior to expiration of the 12 month period since their last exam (T2 or otherwise). PTARS automates the recall (notification) process by printing initial recall letters (Recall 1) and, if necessary, up to three follow-up letters (Recall 2 to Recall 4) to members.

The following topics are covered in this chapter:

- Printing recalls
- Printing recall lists
- Viewing/editing recall dates

The Recalls Menu is accessed by selecting the "Recalls" option from the Main Menu. As shown in Figure 10, three options are available from the Recalls Menu. Each of these options will be discussed in detail in this chapter.

```
01/28/92 12:00:00 am
PTARS RECALLS MENU
<F1> for help
<Alt+F1> for functions
0. Exit to main menu
1. Print recalls
2. pRint most recent recall list
3. View/edit recall dates
select : : 
```

Figure 10. Recalls Menu.

Printing recalls

Select "Print recalls" from the Recalls Menu to immediately start printing recall letters. Note that PTARS always backs-up the current MEMBERS.DBF to the hard disk prior to beginning its print routine. Also, note that prior to printing something, PTARS always presents a "Check the printer" notification. (See Figure 11.) You are also given the option to abort the print job. It is particularly important to heed this notification prior to printing recalls since the printing volume can be over 200 pages during this process and the print job can last over 45 minutes. Moreover, as discussed below, recall dates are inserted into the Members database. Any disruption of this process is problematic.

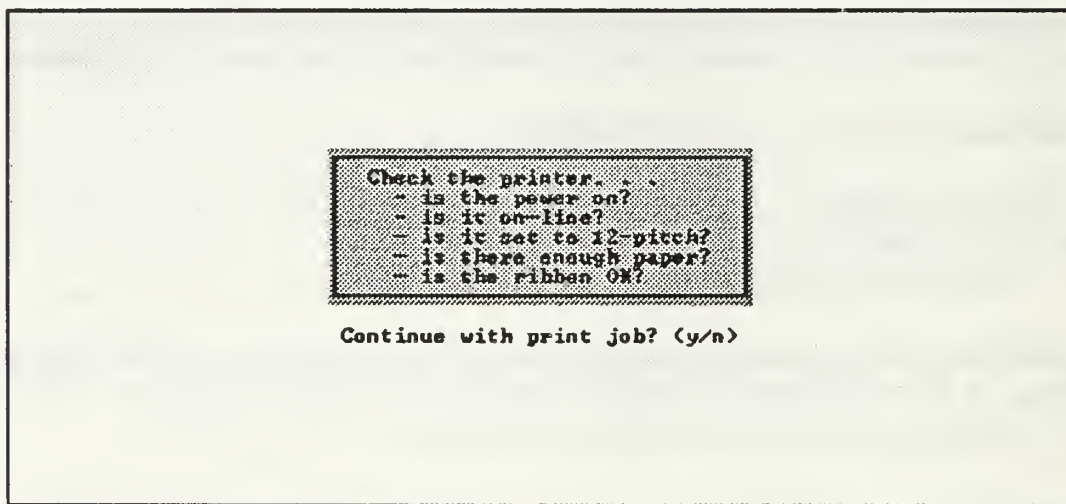


Figure 11. "Check the printer" notification.

It is important that recalls be printed at approximately the same time every month (e.g., the last day of the month or the first day of the month). This will provide consistency in the intervals that members receive follow-up letters, should they be necessary.

When you print recalls, all recall letters are printed and recall letter dates are inserted into MEMBERS.DBF. (Note: The current MEMBERS.DBF is backed-up to the hard disk before printing.) "Print Recalls" also creates a file for each recall letter category which lists members for whom a recall letter is printed (Recall1.lst to Recall4.lst). The previous recall list files are saved with a .BAK extension should they need to be examined from DOS. The logic of recall printing is described following the important section below.

IMPORTANT - The recall letter printing module automatically inserts a new recall letter date into the Members database when a recall letter is printed. It also creates files (RECALL1.LST to RECALL4.LST) containing SSNs and names of members for whom a recall letter was printed. If a printer malfunction occurs or the print job is aborted for some reason, it will be necessary to compare the file listings of the most recent recall letters against the physically printed letters. **Members who are on the file listing, but for whom there is no useable printed recall letter, must have the new recall letter date deleted before the program can print a replacement recall letter.** This is because the printing module checks the existing recall dates to determine if an appropriate recall letter has already been printed.

If for some reason none or relatively few usable recall letters are printed (e.g., the printer was not turned on or there was an early paper jam), you may want to consider restoring the hard disk backup that was created just prior to printing the recalls and starting over. None of the new recall dates will exist on the backup and you can fix the printer and start fresh. See "Restoring backup(s)" in chapter 6. The logic of the recall process is described below:

- Recall 1** Recall 1 is triggered after at least 10 full months + 1 day have transpired since the member's last T2 exam. Prints a memo to the member and records the print date as Recall 1 date.
- Recall 2** Recall 2 is triggered after at least 11 full months + 1 day have transpired since the member's last T2 exam, provided that Recall 1 date is in the database and that at least 25 days have transpired since Recall 1. Prints a memo to the member and records the print date as Recall 2 date.
- Recall 3** Recall 3 is triggered after at least 12 full months + 1 day have transpired since the member's last T2 exam, provided that Recall 2 date is in the database and that at least 25 days have transpired since Recall 2. Prints a letter to the member and records the print date as Recall 3 date.
- Recall 4** Recall 4 is triggered after at least 13 full months + 1 day have transpired since the member's last T2 exam, provided that Recall 3 date is in the database and that at least 25 days have transpired since Recall 3. Prints the letter to the member's superior (i.e., Curriculum Officer for students or to Activity POC for non-students) and records the print date as Recall 4 date.

Example recall letters 1 through 4 are shown in Figures 11 through 14 on the following three pages. Note that the text of Recall 4 indicates that Recall 3 is included as an enclosure. Thus, when routing Recall 4 letters a copy of Recall 3 should be attached. Copies of recall letters can be made by printing from double-copy paper, or alternatively, Xerox copies of just letters 3 and 4 can be made before routing them. The volume of these two letters is historically very low.

1 December 1991

MEMORANDUM (First Reminder)

From: Director, Branch Dental Clinic, Monterey
To: ENS Dandelion Wine, USN, 023-12-3122, NPS STUDENT (SMC 1002)

Subj: ANNUAL DENTAL EXAMINATION

Ref: (a) SECNAVINST 6600.1C
(b) AR 40-35
(c) AF MAN 30-130
(d) COMDTINST M6000.1B

1. References (a) through (d) require that all personnel receive an annual dental examination. Your record indicates that you will be due for an examination next month.
2. Please schedule an appointment with the Dental Clinic in person or by calling 646-2477/2478 at your earliest convenience.
3. If you have had a dental exam within the past 90 days, please contact the dental clinic so that we may update your record. If you have already made an appointment, please disregard this notice.

R. C. TERHUNE

Figure 11. Example Recall 1 memorandum.

1 December 1991

MEMORANDUM (Second Reminder)

From: Director, Branch Dental Clinic, Monterey
To: LCDR Robert O. Bloch, USN, 076-35-3746, NPS STUDENT (SMC 1230)

Subj: ANNUAL DENTAL EXAMINATION

Ref: (a) SECNAVINST 6600.1C
(b) AR 40-35
(c) AF MAN 30-130
(d) COMDTINST M6000.1B

1. References (a) through (d) require that all personnel receive an annual dental examination. Your record indicates that you will be due for an examination this month.
2. Please schedule an appointment with the Dental Clinic in person or by calling 646-2477/2478 within 10 days of receiving this notice.
3. If you have had a dental exam within the past 90 days, please contact the dental clinic so that we may update your record. If you have already made an appointment, please disregard this notice.

R. C. TERHUNE

Figure 12. Example Recall 2 memorandum.

BRANCH DENTAL CLINIC
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943-5100

1 December 1991

From: Director, Branch Dental Clinic, Monterey
To: LT Antoine R. Andrews, USN, 012-12-1212, NDCLB

Subj: ANNUAL DENTAL EXAMINATION DELINQUENCY NOTIFICATION

Ref: (a) SECNAVINST 6600.1C
(b) AR 40-35
(c) AF MAN 30-130
(d) COMDTINST M6000.1B

1. References (a) through (d) require that all active duty military personnel receive a comprehensive dental examination at least once each 12 months.
2. A review of your dental record indicates that your last dental examination was conducted in November, 1990.
3. This facility attempts to assist each member by sending out computerized reminders when their annual examination is due. This was done in your case on 1 October, 1991 and 2 November, 1991 and you failed to respond.
4. It is my responsibility to ensure adherence to the provisions of the references. I am therefore informing you that your annual dental examination must be accomplished prior to 1 January, 1992. Failure to comply will result in further action.
5. You may schedule an examination in person or by calling extension 2477/2478. If you have already made an appointment, please call to confirm it.

R. C. TERHUNE

Figure 13. Example Recall 3 letter.

BRANCH DENTAL CLINIC
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943-5100

1 December 1991

From: Director, Branch Dental Clinic, Monterey
To: Curriculum Officer, Operations Analysis (Code 30)

Subj: MAJOR Larry B. Herman, USAF, 256-98-6582

Encl: (1) Copy of my ltr dtd 1 November, 1991

Ref: (a) SECNAVINST 6600.1C
(b) AR 40-35
(c) AF MAN 30-130
(d) COMDTINST M6000.1B

1. Per references (a) through (d), all active duty military personnel are required to have an annual dental examination. The Branch Dental Clinic, Naval Postgraduate School, contacts individuals requiring examination by sending them a recall notice via the mail. Dental records of personnel that do not respond and exceed the one year limit are marked accordingly and then another recall notice is sent.

2. MAJOR Herman was sent both recall notices and after failing to respond was sent enclosure (1). He/She once again has failed to respond and I must now assume that he/she does not intend to comply with the references.

3. It is requested that MAJOR Herman be appropriately counseled and directed to call extension 2477/2478 to schedule his/her annual dental examination. If you have any questions please feel free to call me at any time.

R. C. TERHUNE

Figure 14. Example Recall 4 letter.

Printing recall lists

Select "pRint most recent recall list" from the Recalls Menu. This option lists (to the printer only) the most recent recall letter information. (The same information is listed to the screen during the printing of the recall letters.) Use this option in the event of a printer malfunction when printing recall letters to compare physical letters against what the program "thinks" it printed. Popup options are presented to select which listing to print. Figure 15 depicts an example listing of Recall 3.

Listing of most recent Recall 3 letters. Created 01/23/92 at 12:00.				
SSN	Last Name	First Name	MI	Last T2
012-12-1212	Andrews	Antoine	R	07/14/90
089-64-3585	Morrison	Larry	R	02/17/89
123-92-9292	Alexander	Hamilton	A	07/12/90
133-21-3838	Zamfir	Jonathan	L	07/12/90
145-89-4509	Lane	Lois	A	04/12/90
149-34-9321	Connors	Jimmy	P	06/14/89
234-58-9234	Delbert	Arnold		07/12/90
282-38-2881	Cricket	Jiminy		07/28/90
283-82-3843	Dean	Larry	X	07/30/90
336-29-3121	Maples	Veronica	S	12/25/89
342-34-5245	Tillerman	Teaforthe		09/01/90
345-21-6587	Rogers	Maybelle	T	12/11/89
345-92-0394	Newman	Alfred	E	04/21/90
383-83-8383	Name	New		07/12/90
408-45-9084	Stevenson	Robert	L	04/21/89
427-84-8320	Diller	Phyllis		02/19/90
489-43-8438	Bell	Dabney		08/12/90
494-59-3493	Dillo	Arma	A	07/12/90
.
.
.

Figure 15. Example listing of Recall 3.

Viewing/editing recall dates

The "View/edit recall dates" option of the Recalls Menu provides a means for viewing recall letter dates for multiple records and for accessing individual records for recall letter date editing. This facility should be used in conjunction with the previously discussed recall listings in the event of a printer malfunction when printing recall letters. The sub-menu options of the View Recalls screen shown in Figure 16 are the same as the like-named options discussed in Chapter 3 for the Delete/view screen. Since recall dates are a subset of the fields in the Members database, records can not be deleted using View Recalls.

<F1> for help		VIEW RECALL DATES				11/28/92 12:00:00 am	
Record#	SSN	LAST NAME, FI	RECALL_1	RECALL_2	RECALL_3	RECALL_4	
1	000-00-0002	Herman, E	11/01/90	12/02/90	01/10/91	10/12/91	
2	001-00-0003	Miserables, L					
3	012-12-1212	Andrews, A	09/03/91	10/04/91			
4	012-93-8475	Adams, J	07/18/90	08/18/90	09/18/90	10/12/91	
5	022-20-0000	Marcos, I					
6	023-12-3122	Wine, D					
7	039-39-2828	Lincoln, M					
8	076-35-3746	Bloch, R	10/04/91				
9	083-82-7827	Mathews, M					
10	089-64-3585	Morrison, L	09/16/90	10/04/91			
11	102-20-0000	Mastroiani, M					
12	109-28-3746	Laverne, S					
13	123-45-6789	Doherty, J	11/21/91				
14	123-58-9213	Madison, J	05/18/91	06/18/91	07/19/91	10/04/91	
15	123-92-9292	Alexander, H	09/16/91	10/12/91			
16	133-21-3838	Zamfir, J	09/16/91	10/12/91			
17	134-15-6789	Sullivan, M	06/12/91	07/12/91	08/12/91	10/04/91	
18	138-38-3838	Hears, R					

VIEW RECALLS: <E>dit <F>ind <G>oto <M>ode <Arrows> <PgDn> <PgUp> <Return>

Figure 16. View Recalls screen.

As discussed previously, the purpose of editing recall letter dates is to enable PTARS to print a replacement recall letter. If a recall letter date is present for a given recall letter, the program will only be able to print the *next* letter when the eligibility date for the *next* recall letter arrives. To reprint a letter, the recall letter date *must* be deleted *and* there *must not* be a subsequent recall letter date present. If this sounds confusing, reread the previous coverage of "Printing Recalls".

To edit a member's recall dates, press {E}. The current row of the display will be highlighted and placed into edit mode. Use normal editing and movement keys to edit the date(s). Note that edited dates are checked for chronological consistency as well as general date validity (i.e., can not be later than the current date, must have a prior recall, can not be missing a recall between recalls, values must be chronologically correct for existent recalls).

Reports

This chapter discusses the various reports available in PTARS and provides several example figures to preview the look of the reports. The Reports Menu, shown in Figure 17, is accessed from the Main Menu by pressing {P}. The Operational Readiness Report is available to both the screen and the printer. The other reports (rosters) are sent to the printer only.

```

11/29/92 12:00:00 am
PTARS REPORTS MENU

<F1> for help
<Alt+F1> for functions

0. Exit to main menu
1. Operational readiness
2. Members <all>
3. members by Class
4. members by UIC <all>
5. members by Pano status
6. Activities
7. cuRriculum

select : :
  
```

Figure 17. Reports Menu.

Operational readiness

The Operational Readiness Report provides counts and percentages of members in each of the dental CLASS categories. The report is initially displayed to the screen and you are given the option of printing it. Operational Readiness is defined as the percentage of all members served by the clinic who are classified as CLASS 1 or 2. As can be seen in Figure 18, the Operational Readiness percentage is a simple summation of the CLASS 1 and CLASS 2 percentages.

BRANCH DENTAL CLINIC, MONTEREY OPERATIONAL READINESS REPORT All Members					January 28, 1992
CLASS CATEGORY:	Class 1	Class 2	Class 3	Class 4	TOTAL
MEMBER COUNT:	1152	566	111	91	1920
PERCENT OF TOTAL:	60%	29%	5.8%	4.7%	100%
OPERATIONAL READINESS:	89%				
PANO CATEGORY:	Green	Red	Yellow		TOTAL
PANO COUNT:	1853	21	46		1920
PERCENT OF TOTAL:	97%	1.1%	1.9%		100%
Print this report? (y/n)					

Figure 18. Operational Readiness Report to screen.

Also included in the report are counts and percentages of members whose Pano X-rays are in a given status. Three Pano status categories exist and are designated by standard color designations:

GRN (Green)	Pano is accepted and on-file
RED	Pano has been duplicated and forwarded
YLW (Yellow)	Pano is not on-file and has not been duplicated and forwarded

Rosters

The remaining reports available from the Reports Menu are basically rosters sorted on various fields of interest. After selecting any of the Members reports a popup will offer a selection of whether to list members by SSN or alphabetically. If printing Members by dental CLASS, a popup will allow selection of a specific CLASS or all members. If printing Members by Pano status, a popup will allow selection of a specific status or all members. Figure 19 provides an example roster of Members listed by SSN that could be printed by selecting option 2, "Members (all)", from the Reports Menu.

Selections 6 and 7 from the Reports Menu print complete rosters of the Activities and the Curriculums contained in their respective PTARS databases.

Periodic comparison of Member rosters against data from both PSD and the Registrar will help keep member data up-to-date. Current listings of the Curriculums at NPS should also be obtained from the Registrar so that the Curriculum database can be kept up-to-date.

FOR OFFICIAL USE ONLY		BRANCH DENTAL CLINIC MONTEREY Member Listing by SSN				January 28, 1992		
SSN	NAME	RANK	SERVICE BRANCH	UIC	SMC/ CODE	LAST T2 EXAM	CLASS	PANO STATUS
000-00-0002	Merman, Ethel	LT	USN	63134	1000	03/21/89	4	GRN
001-00-0003	Miserables, Les	LT	USN	45210		03/21/91	1	GRN
012-12-1212	Andrews, Antoine R.	LT	USN	35728		07/14/90	4	GRN
012-93-8475	Adams, John Q.	ENS	USN	31405	1280	07/12/89	4	YLW
022-20-0000	Marcos, Imelda	CAPT	USA	TRAC		09/12/91	1	RED
023-12-3122	Wine, Dandelion	ENS	USN	31405	1002	07/30/90	4	GRN
039-39-2828	Lincoln, Mark	ENS	USN	31405	1010	11/17/90	4	GRN
076-35-3746	Bloch, Robert O.	LCDR	USN	31405	1230	01/05/90	4	YLW
083-82-7827	Mathews, Mark M.	LTJG	USN	35728		04/12/91	1	YLW
089-64-3585	Morrison, Larry R.	LTJG	USN	31405	1343	02/17/89	4	RED
102-20-0000	Mastroiani, Marcello O.	LT	USN	31405	2030	09/12/91	1	GRN
109-28-3746	Laverne, Shirley	DT2	USN	35728		07/30/91	4	GRN
123-45-6789	Doherty, Janet I.	LT	USN	31405	1001	11/21/90	4	GRN
.
.
.
568-46-4321	Johnson, Emily T.	YN3	USN	43073		06/03/91	1	GRN
571-56-3636	Conseco, Jose F.	ENS	USN	31405	1776	07/12/90	4	GRN
574-84-3823	Than, Smaller X.	LCDR	USN	31405	2312	07/12/91	1	GRN

Page: X

Page: X

Figure 19. Members (all) roster sorted by SSN.

Utilities

This Chapter explains the various utilities included with PTARS that support proper maintenance of the databases. The Utilities Menu is accessed by pressing {U} from the Main Menu and is shown in Figure 20.

It contains the following sections:

- Backup utilities
- Changing the password
- Packing the database(s)
- Changing the date or time
- Selecting the default printer

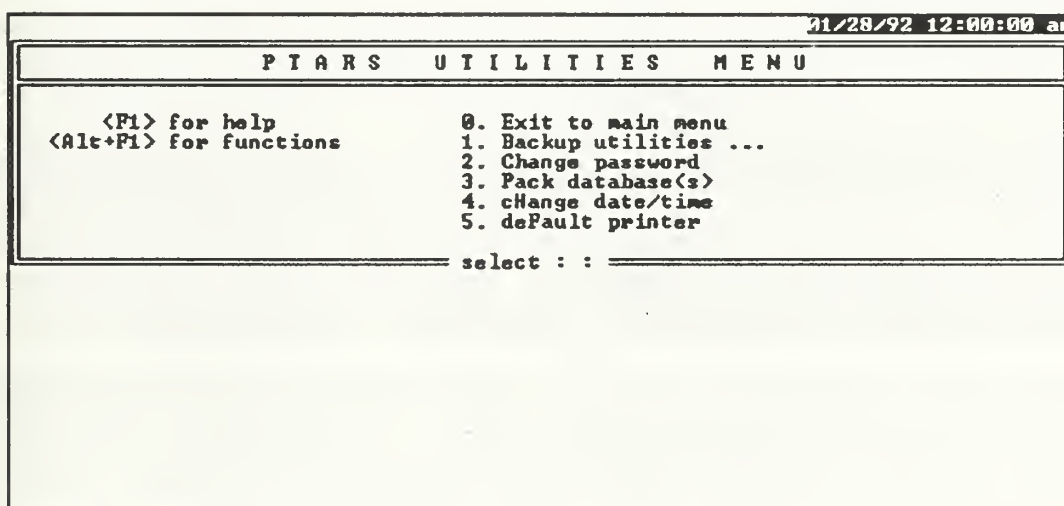


Figure 20. Utilities Menu.

Backup utilities

The backup utilities are a collection of utilities related to backing-up and restoring the four database files MEMBERS.DBF, ACTIVITY.DBF, CURRICUL.DBF, and

DIRECTOR.DBF. The Backup Utilities Menu, shown in Figure 21, is accessed from the Utilities Menu by pressing {B}. Each of the menu selections will be discussed in the sub-sections below.

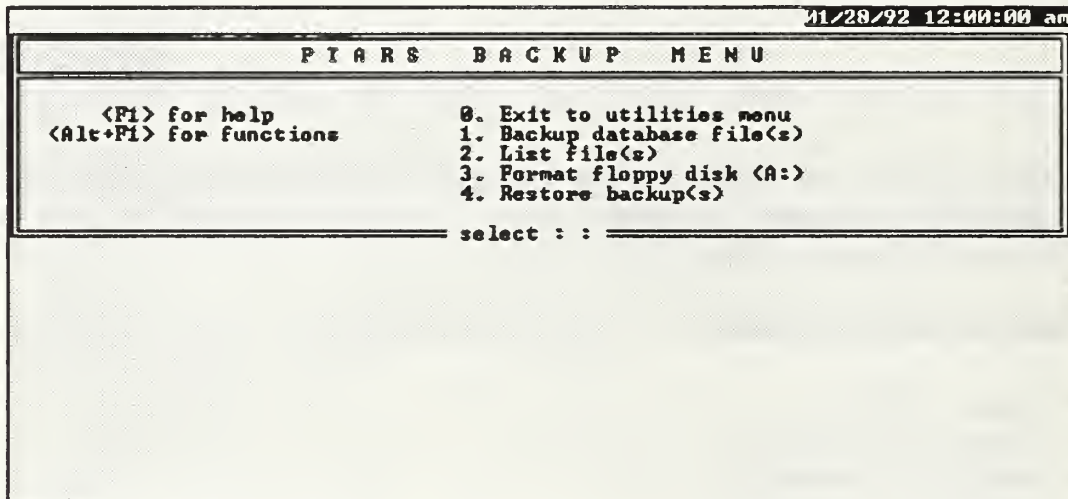


Figure 21. Backup Utilities Menu.

Backing-up database(s)

When you first select Backup, a popup will appear allowing you to select whether you want to back-up to the hard disk or the floppy disk in drive A. Next, another popup appears to let you select which database file(s) (i.e., MEMBERS.DBF, ACTIVITY.DBF, CURRICUL.DBF, DIRECTOR.DBF, or all) to back-up. Once your selection is made, Backup copies the selected file(s) to the destination drive. Backing-up to a floppy keeps a reserve copy of the data that can be restored in case something happens to the computer, hard disk, or the data. Backing-up to the hard disk is convenient for short-term backups, but is *not* sufficient for best reliability. Note that the PTARS program presents the option to back-up the databases to the hard disk prior to quitting a session.

Your data *should* be backed up to a floppy disk weekly and immediately following input or editing sessions involving many records. It is a good idea to keep two or three backup floppies in rotation—writing over the oldest backup each time. *Always* label your backups to floppy disk with the file names and their creation dates. This will help you to identify your backups later if you need to restore them. Hint: use a pencil to label your backups; you can use several floppy disks over and over again by erasing and writing the new information.

Remember, there is only one way to ensure the safety of your data: rigorous adherence to a regular program of backing-up.

Listing files

A popup menu allows selecting the hard disk PTARS subdirectory or floppy disk A: for listing files. Either just database files can be displayed or all files can be displayed. When database files are displayed the following information is included: file name, number of records, last update, file size, total bytes in database files, and bytes remaining on the drive. When all files are displayed, file names are listed and total bytes used in the files and bytes remaining on the drive are presented.

This utility is useful for identifying files that might already exist on a diskette that will be used for backups.

Formatting a floppy disk

Formats a 360Kbyte or a 1.2Mbyte floppy disk (5 1/4") placed in drive A. A popup presents three options:

360K --> 360K	Formats from a 360K capacity drive to a 360K floppy
1.2M --> 360K	Formats from a 1.2M capacity drive to a 360K floppy
1.2M --> 1.2M	Formats from a 1.2M capacity drive to a 1.2M floppy

The first number indicates the actual drive-type. For example, your machine may only be capable of formatting 360K floppy disks, as in the first option. The second number indicates the floppy disk formatted capacity. A new floppy disk must be formatted so that the Disk Operating System (DOS) can read and write data to it.

Restoring backup(s)

When you select "Restore backup(s)", a popup enables selectively replacing database file(s) with backups from the hard disk or a floppy disk.

At the end of every session with PTARS you are presented with the option to backup the databases to the hard disk. If you choose to do so, four backup database files, MEM_BU.DBF, ACT_BU.DBF, CUR_BU.DBF, and DIR_BU.DBF are created in the PTARS subdirectory of your hard drive. These files can be restored (either singly or together) to MEMBERS.DBF, ACTIVITY.DBF, CURRICUL.DBF, and DIRECTOR.DBF, respectively. The restored backups overwrite the current database file(s).

Note that backing-up to the hard drive does not protect your data from hard drive or computer failure since the backups reside on the same machine as the original data. The feature is useful, however, if your original data becomes corrupted for some reason but your backups are still OK. In addition, it may be useful in the event you have experienced a printer malfunction (e.g., your printer ribbon gave up the ghost) and you have many unusable recall letters. Rather than editing recall dates and printing again,

it may be advantageous to restore the backup of MEMBERS.DBF (which PTARS always makes before printing recalls) and start over.

A final method of restoring any database is to manually copy the file using DOS commands. This method should never be necessary since the capability is built into PTARS. If for some reason you should need to manually restore a *.DBF file, be sure that any like-named compound index file (*.CDX) is erased (e.g., from the DOS prompt: `del c:\ptars\members.cdx`) This is because a unique index file is created and updated by PTARS for each database. If the index file does not "belong" to the specific version of a database, PTARS will not perform properly and will give an error notification.

Changing the password

You can change the current password to a new password (it must have 6 characters). Make sure that you *remember* the new password. If you ever forget your new password, copy the file NPS_MISC.DBF from disk 3 of your *backup copies* of the installation disks to the sub-directory \PTARS (e.g., copy `a:\nps_misc.dbf c:\ptars`). The original password is "zyxabc". This default password should be changed immediately after you install PTARS. (If you can read it here, so can someone else.) Note that the password is encrypted in the file NPS_MISC.DBF and cannot be deciphered if it is forgotten.

Figure 22 shows the screen for changing the password. As you type your new password, a dot will appear for each character typed. As shown in the figure, to verify that you typed what you thought you typed, PTARS prompts for a second entry of your new password. If the two entries do not match, the password change will be aborted.

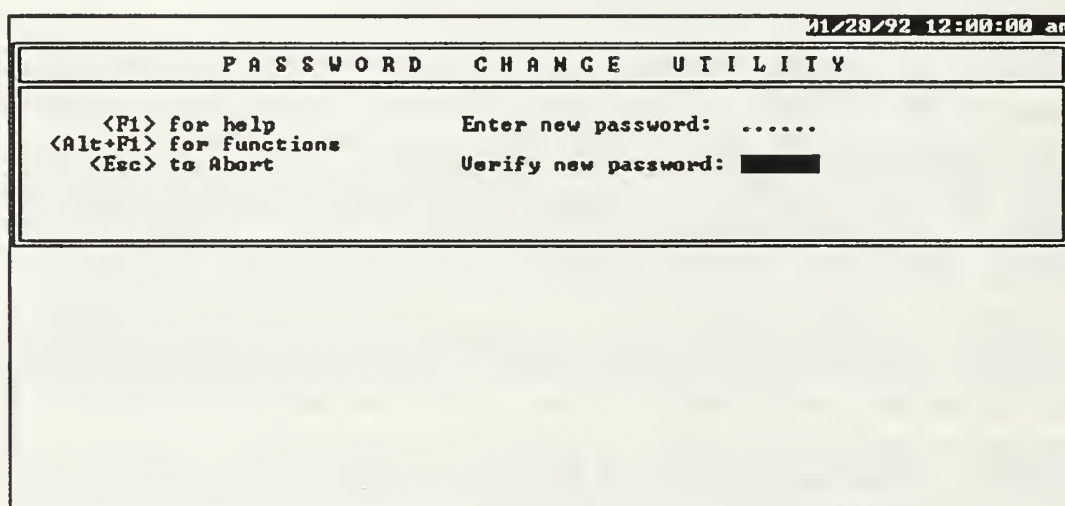


Figure 22. Password change screen.

For effective security it is a good idea to periodically change your password. If an unauthorized individual inadvertently (or even deliberately) changes or damages your data, it could be a catastrophe. Regarding security, just think about having to re-enter over 1900 records!

Packing the database(s)

Packing the database(s) *permanently* deletes records "marked" for deletion from one or all of the three primary databases: MEMBERS.DBF, ACTIVITY.DBF, and CURRICUL.DBF. It also physically sorts the databases. MEMBERS.DBF is sorted in ascending order by SSN; ACTIVITY.DBF is sorted in ascending order by UIC; and CURRICUL.DBF is sorted in ascending order by curriculum number. Packing improves the performance of PTARS by reducing the physical size of the database(s) and reorders the records by the primary key. Note that the effects of packing are *not* "undoable". An informational prompt will appear upon quitting a session when 10% or more of the MEMBER.DBF records are marked for deletion. You should heed the prompt and pack the database (unless you have some good reason not to).

Changing the date or time

After selecting the "Change date or time" option a popup for selecting which to change (date or time) appears. After your selection is made you are prompted to enter the date or time using the example format shown on the screen. The system date or time can also be changed when starting the PTARS program. As part of the opening screen routine the user is prompted to verify the system date and time. If the system date or time displayed is incorrect, enter the correct date or time using the example format shown on the screen.

Selecting the default printer

You should select the default printer before printing anything from PTARS for the first time. After choosing this option from the Utilities Menu, PTARS pops-up two common printer emulations for dot matrix printers: (1) Epson E/F/J/RX/LQ emulation and (2) IBM Proprinter emulation. The emulation you select becomes the default for all subsequent sessions. The Epson emulation is supported by the majority of 9 pin dot matrix printers and PTARS uses it as the initial default. The default printer identifier is stored in a field in the NPS_MISC.DBF file.



PTARS User's Manual

The PTARS (Public Transport Analysis and Reporting System) is a software package designed to assist in the analysis and reporting of public transport data. It is intended for use by transport planners, analysts, and researchers. The system provides a comprehensive set of tools for data processing, analysis, and visualization, enabling users to generate detailed reports and charts from their data. The manual describes the installation, operation, and various features of the PTARS system, including data input/output, analysis modules, and report generation options.

The PTARS system is designed to be user-friendly and flexible, allowing users to tailor the analysis and reporting to their specific needs. It supports a wide range of data formats and provides a variety of analysis options, including route analysis, demand analysis, and performance analysis. The system also includes a robust reporting module that can generate reports in a variety of formats, including HTML, PDF, and print-ready formats.

For more information about the PTARS system, please contact the PTARS support team. The support team is available to provide assistance with installation, operation, and any other issues that may arise. The PTARS system is a powerful tool for public transport analysis and reporting, and we hope that this manual will help you to get the most out of it.

Optimizing PTARS

This appendix identifies several ways that you can optimize the performance of PTARS if you have certain hardware or software capabilities. It contains the following sections:

- Disk defrag/compress
- Memory
- Config.sys
- Pack the database(s)

Disk defrag/compress

The performance of PTARS can be significantly improved if a disk defragment/compression procedure is performed on your hard drive periodically. Over time the database files will become fragmented as records are appended, edited and deleted. This slows down disk reads and writes since each file is no longer one contiguous piece; files can become many pieces scattered all over the disk. Defragment/compression utilities are available commercially.

Memory

PTARS will take advantage of all types of computer memory. If your computer is configured correctly, PTARS' performance will be enhanced. Note that if you change your computer's memory configuration or add a disk cache program, you must re-install PTARS so that it operates optimally.

Personal Computers (PC)s can contain three types of memory: conventional, expanded and extended.

Conventional Memory

All PCs can contain conventional memory (up to 640K). This is the memory that programs typically load into and run in. PTARS requires that you have at least 512K of conventional memory with at least 420K of it free after memory resident programs have been loaded. A minimum of 640K is *strongly* recommended.

Expanded Memory

The 8086 family of microprocessors have a physical address space of 1024K, or 1MB. The first 640K is the conventional memory space discussed above. The remaining 384K is reserved for use by read-only memory (ROM) and hardware device controllers. Also, within this area of memory, a 64K block can be reserved for use by an expanded memory manager which conforms to the Lotus/Intel/Microsoft interface specification (a LIM EMS Memory Manager).

The Expanded Memory Manager (EMM) administers expanded memory as a system resource that can be used by several applications at the same time and services EMS function calls. EMS memory is bank-switched memory that can be larger than the CPU's address space that is mapped into conventional memory via the EMS page frame.

On machines with expanded memory that is LIM 4.0 EMS compatible, PTARS uses the first 64K of expanded memory as "general purpose" memory and any remaining expanded memory to speed file I/O and to cache PTARS code segments.

To check how much EMS is currently being used by PTARS, look in the "About PTARS" box (by pressing <F4> or <Alt+F1>).

If you run on an 80386 or 80486 you're in luck! There are many inexpensive programs that use extended memory to emulate EMS, such as QEMM from Quarterdeck and 386MAX from Qualitas. MS-DOS 5.0 includes EMM386. On a 386 **always** use QEMM, 386Max, or other expanded memory managers. You'll be glad you did!

If you use a non-80386 processor you have several options. First, you could invest in an EMS board. These pieces of hardware, which usually work with both 8086/88 and 80286 processors, include substantial amounts of memory together with driver programs which provide the software interface to the board.

Extended Memory

Extended memory is memory that lies above the 1MB address range. It can be used directly by some operating systems (OS/2 and UNIX), but standard DOS cannot address it without the use of an Extended Memory Specification (XMS) driver, an interface that allows access to memory beyond 640K. Applications using this address space must be running in protected mode.

Extended memory cannot be used directly by PTARS until it is made to act like EMS. How you make extended memory act like expanded memory is dependent on your system, but typically you install a memory manager -- software that provides an EMS style (LIM 4.0) interface to extended memory. Once the extended memory is emulating EMS memory, PTARS will sense that it is there and make good use of it.

Config.sys

The system configuration file, CONFIG.SYS, contains certain commands that are checked and executed when you start up your computer. These commands change your computer's default configuration.

CONFIG.SYS is not a PTARS file. It's a file that DOS uses to establish the working environment. Because PTARS interacts with this environment, you must be sure that certain settings are properly established. Two CONFIG.SYS statements are of immediate importance to PTARS:

BUFFERS The BUFFERS statement contains the number of disk buffers that DOS sets aside in memory when your computer is started. A disk buffer is a block of memory (typically 512 bytes) that DOS uses to hold data when reading and writing from disk. For best performance with PTARS, the CONFIG.SYS file should contain a BUFFERS statement with a number between 20 and 40 (e.g., BUFFERS=30).

FILES The FILES statement sets the number of files that DOS can open and access at one time. This number is directly related to the number of files that PTARS will be able to open. The FILES statement in CONFIG.SYS should always be at least 25 (e.g., FILES=25).

See your DOS manual for complete details on the CONFIG.SYS file and the various statements it can contain.

Pack the database(s)

Packing the databases is covered in Chapter 6.



File definitions

The files listed below (with their definitions) are installed by Setup into the "\PTARS" hard disk subdirectory. These files are essential to the operation of PTARS. Three of the files, FOXPRO.ESL, FOXPRO.ESO, and PTAR.EXE are in compressed form on the installation disks and will not work if copied directly from the floppy disk to your hard drive. All of the other files installed by PTARS are in normal form on the installation disks.

PTARS files

CONFIG.FP	resource pointer file
FOXPRO.ESL	database routines library
FOXPRO.ESO	database routines library
CACHE.COM	extended memory (512K req'd) disk cache utility
NPS_MISC.DBF	contains encrypted password, default printer, backup date
NPS_USER.DBF	contains configuration information
NPS_USER.FPT	memo file for configuration information
PTAR.EXE	PTARS executable program
PTARS.COM	PTARS loader program

NPSDC database files

ACTIVITY.DBF	contains UIC information
CURRICUL.DBF	contains student Curriculum information
DIRECTOR.DBF	contains current Director signature name
MEMBERS.DBF	contains Member information

The following files are created during the operation of PTARS and may or may not be present at any given time:

ACTIVITY.CDX	compound index file for ACTIVITY.DBF
CURRICUL.CDX	compound index file for CURRICUL.DBF
MEMBERS.CDX	compound index file for MEMBERS.DBF
ACT_BU.DBF	hard disk backup of ACTIVITY.DBF
CUR_BU.DBF	hard disk backup of CURRICUL.DBF
DIR_BU.DBF	hard disk backup of DIRECTOR.DBF
MEM_BU.DBF	hard disk backup of MEMBERS.DBF

RECALL1.LST	most recent listing of members receiving recall 1 letter
RECALL2.LST	most recent listing of members receiving recall 2 letter
RECALL3.LST	most recent listing of members receiving recall 3 letter
RECALL4.LST	most recent listing of members receiving recall 4 letter
RECALL1.BAK	previous listing of members receiving recall 1 letter
RECALL2.BAK	previous listing of members receiving recall 2 letter
RECALL3.BAK	previous listing of members receiving recall 3 letter
RECALL4.BAK	previous listing of members receiving recall 4 letter
RELATE1.VUE	PTARS environment file
RELATE2.VUE	PTARS environment file



Database specifications

Members.dbf

<u>Field-name</u>	<u>Type</u>	<u>Length</u>	<u>Usage</u>
SSN	Character	11	Social Security Number -- unique, mandatory, key field
LAST_NAME	Character	23	Last Name -- mandatory
FIRST_NAME	Character	15	First Name -- mandatory
MI	Character	1	Middle Initial -- if available
RANK_RATE	Character	5	Rank or Rate -- mandatory
BRANCH	Character	4	Service Branch -- mandatory, popup list
LAST_T2	Date	8	Last-T2-Exam date -- mandatory
CLASS	Numeric	1	Dental Class -- mandatory, range (1 - 4), PTARS updated
PANO	Character	3	Pano X-ray status -- mandatory, popup list
UIC	Character	5	Unit Identification Code -- mandatory, popup list, linked with ACTIVITY.DBF (used in "To:" line of recall letters to students)
CURR_NUM	Character	3	Curriculum Number -- mandatory for UIC 31405, popup list, linked with CURRICUL.DBF
SMC/CODE	Character	4	Student Mail Center number/Department Code -- if available (used in "To:" line of recall letters)
RECALL_1	Date	8	Recall 1 letter date -- PTARS created, editable
RECALL_2	Date	8	Recall 2 letter date -- PTARS created, editable
RECALL_3	Date	8	Recall 3 letter date -- PTARS created, editable
RECALL_4	Date	8	Recall 4 letter date -- PTARS created, editable

Activity.dbf

<u>Field-name</u>	<u>Type</u>	<u>Length</u>	<u>Usage</u>
UIC	Character	5	Unit Identification Code -- unique, mandatory, key field
ACRONYM	Character	11	Acronym for UIC -- mandatory (used in "To:" line of recall letters 1 - 3)
ACT_NAME	Character	47	UIC Name -- mandatory (used in "To:" line of recall 4 letter)
POC	Character	20	UIC Point of Contact -- mandatory (used in "To:" line of recall 4 letter)

Curricul.dbf

<u>Field-name</u>	<u>Type</u>	<u>Length</u>	<u>Usage</u>
CURR_NUM	Character	3	Curriculum Number -- unique, mandatory, key field
CURR_NAME	Character	46	Curriculum Name -- mandatory (used in "To:" line of recall 4 letter applicable to students)
DEPT_CODE	Character	2	Department Code of Curriculum -- mandatory (used in "To:" line of recall 4 letter applicable to students)
PHONE_NO	Character	4	Curriculum Office Phone Number -- mandatory

Director.dbf

<u>Field-name</u>	<u>Type</u>	<u>Length</u>	<u>Usage</u>
DIRECTOR	Character	20	Director signature -- mandatory (format as per signature line of recall letters)

APPENDIX D: RELATION DEFINITIONS

MEMBER

<u>Item</u>	<u>Type</u>	<u>Length</u>
SSN	Character	11
Last-name	Character	23
First-name	Character	15
MI	Character	1
Rank_rate	Character	5
Branch	Character	4
Last_T2	Date	8
Class	Numeric	1
Pano	Character	3
UIC	Character	5
Curr-num	Character	3
SMC/Code	Character	4
Recall_1	Date	8
Recall_2	Date	8
Recall_3	Date	8
Recall_4	Date	8

ACTIVITY

<u>Item</u>	<u>Type</u>	<u>Length</u>
UIC	Character	5
Acronym	Character	11
Act-name	Character	47
POC	Character	20

CURRICULUM

<u>Item</u>	<u>Type</u>	<u>Length</u>
Curr-num	Character	3
Curr-name	Character	46
Dept_code	Character	2
Phone_no	Character	4

FILELIST.DOC 1 of 1

36

401 tokens are included in this report.

Legend for content symbols:
(blank) reference does not change the variable or field value.
* variable or field is changed in an assignment statement.
! PROCEDURE or FUNCTION statement.
x variable is released.
A array is declared.
G GET or RENU statement changes variable or field.
P variable is declared PUBLIC.
R field is replaced.
U database is used.
V variable is declared PRIVATE.
* variable is referenced in a macro--takes precedence over all others.
7 reference is of unknown type.

File types appear next to tokens that are used as file names. Some of these tokens may also be used in other ways.

ABOUT (procedure in NPS.PROC.PRG)

NPSDC.PRG 212
NPS.PROC.PRG 1791 183 186 235 237x 314

ACRONYM

NPS.PROC.PRG 1350R
NPS.BROW.PRG 322

ACTIVITY

NPS.PROC.PRG 1453
NPS.OPEN.PRG 1100 1700 176
NPS.REPO.PRG 36

ACTIVITY.ACRONYM

NPS.PROC.PRG 1049G 1146

ACTIVITY.ACT_NAME

NPSDC.PRG 275
NPS.PROC.PRG 1048G 1145

ACTIVITY.PDC

NPS.PROC.PRG 1070G 1147

ACTIVITY.UIC

NPSDC.PRG 275
NPS.PROC.PRG 1047G 1144
NPS.BROW.PRG 167

ACT NAME

NPS.PROC.PRG 1349R
NPS.BROW.PRG 322

ACT.UIC

NPS.OPEN.PRG 114
ADISPLAY (procedure in NPS.BROW.PRG)

NPS.BROW.PRG 2511 378

ALL NAME

NPS.REPO.PRG 37

ALL.SSN

NPS.REPO.PRG 38

ALT

NPSDC.PRG 213

BACKUP UTILS

NPS.PROC.PRG 158

BAKFILE

NPS.RECA.PRG 282* 285* 287*
(procedure in NPS.PROC.PRG)

BAR1

NPS.PROC.PRG 259 2741
(procedure in NPS.PROC.PRG)

BAR2

NPS.PROC.PRG 260 2871
(procedure in NPS.PROC.PRG)

BAR3

NPS.PROC.PRG 261 3001
(procedure in NPS.PROC.PRG)

BAR4

NPS.PROC.PRG 242 3131
(procedure in NPS.PROC.PRG)

BDISPLAY

NPS.BROW.PRG 2971 380
(procedure in NPS.BROW.PRG)

BRANCH

NPSDC.PRG 255 258 259 260 261 262 263 264
NPS.PROC.PRG 1126 1326 1414
NPS.BROW.PRG 283

BRANID

NPS.PROC.PRG 1409
(procedure in NPS.BU.PRG)

BU

NPSDC.PRG 54 61 100 400*
NPS.BU.PRG 98 142 145 1731 176 179 180 181 182 183
184 187 310x 348 379 382 389x 393 406

BUF

NPSDC.PRG 444V 453* 454 458* 459 463* 464 467* 468
NPS.OPEN.PRG 43V 38* 72
NPS.RECA.PRG 174V 198* 199 203* 204 208* 209 212* 213 248*
NPS.BU.PRG 269 271* 272 280* 281 283* 284 292* 293 295*
296 303* 304 306* 307 327V 334* 337 341* 342
344* 347 350* 351 640V 682* 683 684* 687 690*
691 694* 695 699* 703 704* 707 710* 711 714*
715 722* 723 725* 729 734* 737 739* 743 750*
751 753* 757 764* 765 767* 771

BUFILE

NPS.PROC.PRG 401 413 415* 418 421* 426 428* 438 439* 443*
NPS.BU.PRG 448 793 802 804* 809 839 856 845 847* 872
902

BU.ALL

NPS.BU.PRG 241 253 3261
(procedure in NPS.BU.PRG)

BU.DISK

NPS.BU.PRG 128V 141* 144* 193 214 247 270 279 282 291
294 302 305

BU.RENU

NPS.BU.PRG 38 791
(procedure in NPS.BU.PRG)

BU.SELECT

NPS.BU.PRG 54 1271 129 132 133 134 137 147x
(procedure in NPS.BU.PRG)

BU.SINGLE

NPS.BU.PRG 272 284 296 307 3471
(procedure in NPS.BU.PRG)

CAL

NPS.PROC.PRG 73 76 77 94x
CALENDAR

NPS.PROC.PRG 77 78
(procedure in NPS.BROW.PRG)

CDISPLAY

NPS.BROW.PRG 3351 382

CHOICE

NPSDC.PRG 58 43 109* 110 111 294 352* 353 355* 356*
361 362* 362 397* 398 399 216 229 230 500 502*
180V 200* 201 202 215 618 621 624 627 670* 671 675
503 504* 615 1603 1604
677 1602* 1603 1604
20V 134* 135 136 147* 148 149
52
55* 59 121 123 127 132 150 155
88 90 92 114 118 131 133 134 148 154
158 204 214 224 227 232 237
48 49 51 51 62 62 117 117 121 121 121
382* 383 385 387 409 413 421 429 447 449
450 464 470 480
53 54 54 54 65 65 94 95 100 100
109 109 119 119 128 128 138 138 146 146
32 33 35 35 39 43 43 49 49 54
54 63 63 67 67 216 281 282
NPS.BU.PRG 40 41 43 43 43 47 51 51 54 54
60 60 64 64 228 229 383 384 588* 607
609
39 40

CLASN

NPS.OPEN.PRG 84

CLASS

NPS.PROC.PRG 1128 1328
NPS.OPEN.PRG 85 85 86 224R
NPS.BROW.PRG 283 74 75 76 282 284 286 290 290
NPS.REPO.PRG 292

CLASSPOP

NPS.REPO.PRG 117 2641 264 269 270 271 272 273 274 275
276 279 296x
(procedure in NPS.REPO.PRG)

CLAS.CTP

NPS.REPO.PRG 39

CLAS.CTS

NPS.REPO.PRG 91

CLAS.NAM

NPS.REPO.PRG 40

CLAS.SSN

NPS.REPO.PRG 41

COLGET

NPS.INTR.PRG 20V 20V 93* 98
299V 324* 330 354

COLSAY

NPS.INTR.PRG 94* 100
299V 325* 341 343

COL_CNTR

NPS.INTR.PRG 19V
(procedure in NPSDC.PRG)

CONFIG.CHECK

NPSDC.PRG 219 2281
(procedure in NPSDC.PRG)

COUNTER

NPS.INTR.PRG 18V 25* 26 28 30 32 34 36 38 40
55* 55 87* 108* 110 111 113 114
NPS.EOLT.PRG 32V
NPS.RECA.PRG 34V 214* 217 218* 218 219

CURRICUL

(database)
NPS.PROC.PRG 1491
NPS.OPEN.PRG 124V 172U 177
NPS.REPO.PRG 42

CURRICUL.CURR_NAME

NPSDC.PRG 281
NPS.PROC.PRG 1084G 1140

CURRICUL.CURR_NUM

NPSDC.PRG 281
NPS.PROC.PRG 1083G 1159 1367R
NPS.BROW.PRG 184

CURRICUL.DEPT_CODE

NPS.PROC.PRG 1085G 1141

CURRICUL.PHONE_NO

NPS.PROC.PRG 1086G 1142

CURRNUM

NPS.PROC.PRG 1488 1492

CURR_NAME

NPS.PROC.PRG 1348R
NPS.BROW.PRG 353

CURR_NUM

NPSDC.PRG 1131 1331
NPS.OPEN.PRG 89 130 159 177
NPS.EDIT.PRG 178R
NPS.BROW.PRG 284 353

CURR_NUMC

NPS.OPEN.PRG 130

CURR_NUMS

NPS.OPEN.PRG 89
(procedure in NPS.UTIL.PRG)

DATE.TIME

NPS.UTIL.PRG 64 1611 163 166 167 168 171 180x
DBFAREA
NPSDC.PRG 57 42 83* 305 308 311 314
NPS.PROC.PRG 695* 697 700 703 706 734 746* 763 776 779

XREF.DOC 2 of 7

KEYWORDS	500	503	143	147	150	MEMBERS_LAST_T2	10326
NPS_PROG.PRG						NPS_PROG.PRG	
KEYCODE						MEMBERS_HI	10296
NPS_PROG.PRG	501V	504+ 505				NPS_PROG.PRG	
NPS_INTR.PRG	20V					MEMBERS_PAND	10346
KEYNAM	33V	123+ 125+ 127+ 134				NPS_PROG.PRG	10306
NPS_EDIT.PRG						MEMBERS_BANK_RATE	
KEYSTROKES						NPS_PROG.PRG	10316
NPS_PROG.PRG	180V	187+ 201 215 229				MEMBERS_RECALL_1	10516
NPS_BROW.PRG	24V	41+ 88 114				NPS_PROG.PRG	99
NPS_RECA.PRG	33V	347+ 383 409				MEMBERS_RECALL_2	10526
LASTPAGE						NPS_PROG.PRG	93
NPS_EDIT.PRG	32V					MEMBERS_RECALL_3	10536
LASTREC						MEMBERS_RECALL_4	10546
NPS_PROG.PRG	59	44 122				MEMBERS_SVC	10446 10486
NPS_PROG.PRG	710+					MEMBERS_SSN	
NPS_OPEN.PRG	51					NPS_PROG.PRG	10246
NPS_APPE.PRG	7	115 126 144+ 164				MEMBERS_UTILC	10356 1036
NPS_EDIT.PRG	92					MEMBERS_UTILC	
NPS_BROW.PRG	149					MEMBERS_UTILC	
NPS_RECA.PRG	445					MEMBERS_UTILC	
NPS_UTIL.PRG	39					MEMBERS_UTILC	
NPS_BU.PRG	47					MEMBERS_UTILC	
LASTREQUM						MEMBERS_UTILC	
NPS_PROG.PRG	580	585 588 608 619 628				MEMBERS_UTILC	
LAST_NAME						MEMBERS_UTILC	
NPS_PROG.PRG	1122	1322				MEMBERS_UTILC	
NPS_OPEN.PRG	84	86 91				MEMBERS_UTILC	
NPS_BROW.PRG	281					MEMBERS_UTILC	
NPS_RECA.PRG	522	548				MEMBERS_UTILC	
LAST_T2						MEMBERS_UTILC	
NPS_PROG.PRG	1127	1327				MEMBERS_UTILC	
NPS_OPEN.PRG	225					MEMBERS_UTILC	
NPS_EDIT.PRG	195					MEMBERS_UTILC	
NPS_BROW.PRG	283					MEMBERS_UTILC	
NPS_RECA.PRG	85	98 104				MEMBERS_UTILC	
LEFTAROW						MEMBERS_UTILC	
NPS_BROW.PRG	25V	39+ 41 227				MEMBERS_UTILC	
NPS_RECA.PRG	33V					MEMBERS_UTILC	
LISTPOP						MEMBERS_UTILC	
NPS_PROG.PRG	107	114 126 135 211+ 212 215 216 217 220				MEMBERS_UTILC	
NPS_REPO.PRG	253+					MEMBERS_UTILC	
LISTRECS						MEMBERS_UTILC	
NPS_BROW.PRG	253	254 281 283 284 299 300 322 324 337				MEMBERS_UTILC	
NPS_RECA.PRG	338	353				MEMBERS_UTILC	
NPS_RECA.PRG	502	503 522				MEMBERS_UTILC	
LISTTYPE						MEMBERS_UTILC	
NPS_REPO.PRG	34+	104+ 115+ 125+ 134+ 224 227 230 233 239				MEMBERS_UTILC	
NPS_REPO.PRG	242	245 248				MEMBERS_UTILC	
LIST_DISK						MEMBERS_UTILC	
NPS_BU.PRG	451V	444+ 467+ 496 502 514 520				MEMBERS_UTILC	
LIST_FILES						MEMBERS_UTILC	
NPS_BU.PRG	465	468 482+ 483 484 487 488 491 531+				MEMBERS_UTILC	
LIST_SELECT						MEMBERS_UTILC	
NPS_BU.PRG	59	450+ 452 455 456 457 460 470+				MEMBERS_UTILC	
LISTFILE						MEMBERS_UTILC	
NPS_RECA.PRG	281+	287+ 289+				MEMBERS_UTILC	
MACRONUM						MEMBERS_UTILC	
NPS_PROG.PRG	71	76				MEMBERS_UTILC	
NPS_PROG.PRG	1146	12446 1350				MEMBERS_UTILC	
MACT_NAME						MEMBERS_UTILC	
NPS_PROG.PRG	71	76				MEMBERS_UTILC	
NPS_PROG.PRG	1145	12616 1262 1349				MEMBERS_UTILC	
NPS_APPE.PRG	257					MEMBERS_UTILC	
MAIN						MEMBERS_UTILC	
NPS_PROG.PRG	125					MEMBERS_UTILC	
MAINCHOICE						MEMBERS_UTILC	
NPS_PROG.PRG	58	63 85+ 89 92 122 126 130 135 139				MEMBERS_UTILC	

[illegible]

XREF.DOC 5 of 7


```

1  ; *****
2  ; *****
3  ; *****
4  ; *****
5  ; *****
6  ; *****
7  ; *****
8  ; *****
9  ; *****
10 ; *****
11 ; *****
12 ; *****
13 ; *****
14 ; *****
15 ; *****
16 ; *****
17 ; *****
18 ; *****
19 ; *****
20 ; *****
21 ; *****
22 ; *****
23 ; *****
24 ; *****
25 ; *****
26 ; *****
27 ; *****
28 ; *****
29 ; *****
30 ; *****
31 ; *****
32 ; *****
33 ; *****
34 ; *****
35 ; *****
36 ; *****
37 ; *****
38 ; *****
39 ; *****
40 ; *****
41 ; *****
42 ; *****
43 ; *****
44 ; *****
45 ; *****
46 ; *****
47 ; *****
48 ; *****
49 ; *****
50 ; *****
51 ; *****
52 ; *****
53 ; *****
54 ; *****
55 ; *****
56 ; *****
57 ; *****
58 ; *****
59 ; *****
60 ; *****
61 ; *****
62 ; *****
63 ; *****
64 ; *****
65 ; *****
66 ; *****
67 ; *****
68 ; *****
69 ; *****
70 ; *****
71 ; *****
72 ; *****
73 ; *****
74 ; *****
75 ; *****
76 ; *****
77 ; *****
78 ; *****
79 ; *****
80 ; *****
81 ; *****
82 ; *****
83 ; *****
84 ; *****
85 ; *****
86 ; *****
87 ; *****
88 ; *****

```

```

89 DO main_menu WITH mainchoice
90 SET COLOR TO W/N,W/N
91 DO CASE
92 CASE mainchoice = 0 "returnkey"
93 SET COLOR TO W/N,W/N
94 IF (1 no_dbf) && if no database files are missing do quit sequen
95 "ce
96 DO promptbu
97 "ling
98 DO pack_check
99 IF (bu hard_bu = 11) && if backing up to floppy before quitting
100 ELSE
101 ELSE
102 ELSE
103 ELSE
104 ELSE
105 ELSE
106 ELSE
107 ELSE
108 ELSE
109 ELSE
110 ELSE
111 ELSE
112 ELSE
113 ELSE
114 ELSE
115 ELSE
116 ELSE
117 ELSE
118 ELSE
119 ELSE
120 ELSE
121 ELSE
122 ELSE
123 ELSE
124 ELSE
125 ELSE
126 ELSE
127 ELSE
128 ELSE
129 ELSE
130 ELSE
131 ELSE
132 ELSE
133 ELSE
134 ELSE
135 ELSE
136 ELSE
137 ELSE
138 ELSE
139 ELSE
140 ELSE
141 ELSE
142 ELSE
143 ELSE
144 ELSE
145 ELSE
146 ELSE
147 ELSE
148 ELSE
149 ELSE
150 ELSE
151 ELSE
152 ELSE
153 ELSE
154 ELSE
155 ELSE
156 ELSE
157 ELSE
158 ELSE
159 ELSE
160 ELSE
161 ELSE
162 ELSE
163 ELSE
164 ELSE
165 ELSE
166 ELSE
167 ELSE
168 ELSE
169 ELSE
170 ELSE
171 ELSE
172 ELSE
173 ELSE

```

```

174 SET TALK OFF
175 SET DEFAULT TO SYS(2004)
176 SET PATH TO SYS(2004)
177 IF NOT FILE("NPS_USER.DBF")
178 SET COLOR TO W/N
179 77 CHR(7)
180 WAIT "NPS_USER.DBF file not found. Program exiting..." WINDOW TIMEOUT
181 2 CLEAR
182 0.0 SAY "NPS_USER.DBF not found. Check User's Manual." COLOR W/N
183 QUIT
184 ENDIF
185 SET RESOURCE TO nos_user
186 SET COLOR SET TO DEFAULT
187 SET COLOR OF SCHEME 1 TO W/N
188 SET CURSOR OFF
189 CLEAR
190 SET HELP TO nos_hlp
191 SET MOUSE OFF
192 SET SYSTEM OFF
193 SET STATUS OFF
194 SET BELL OFF
195 SET CARRY OFF
196 SET MENUS OFF
197 SET DOHISTORY OFF
198 SET SAFETY OFF
199 SET ESCAPE OFF
200 SET SCOREBOARD OFF
201 SET CLOCK OFF
202 SET FUNCTION "2" TO
203 SET FUNCTION "3" TO
204 SET FUNCTION "4" TO
205 SET FUNCTION "5" TO
206 SET FUNCTION "6" TO
207 SET FUNCTION "7" TO
208 SET FUNCTION "8" TO
209 SET FUNCTION "9" TO
210 ON KEY LABEL 42 DO pascal
211 ON KEY LABEL 43 DO pascal
212 ON KEY LABEL 44 DO pascal
213 ON KEY LABEL alt+f1 DO func_list
214 m ---Set Capslock off
215 m ---Set Capslock (F.)
216 m ---Set Mouselock off
217 m ---Set Mouselock (F.)
218 m ---Check config.sys files available
219 DO config_check
220 RETURN
221 ; *****
222 ; *****
223 ; *****
224 ; *****
225 ; *****
226 ; *****
227 ; *****
228 ; *****
229 ; *****
230 ; *****
231 ; *****
232 ; *****
233 ; *****
234 ; *****
235 ; *****
236 ; *****
237 ; *****
238 ; *****
239 ; *****
240 ; *****
241 ; *****
242 ; *****
243 ; *****
244 ; *****
245 ; *****
246 ; *****
247 ; *****
248 ; *****
249 ; *****
250 ; *****
251 ; *****
252 ; *****
253 ; *****
254 ; *****
255 ; *****
256 ; *****
257 ; *****
258 ; *****
259 ; *****
260 ; *****

```

```

241 DEFINE BAR 4 OF BRANCH PROMPT "USAVE"
242 DEFINE BAR 5 OF BRANCH PROMPT "USVCC"
243 DEFINE BAR 6 OF BRANCH PROMPT "MXCDA"
244 ON SELECTION POPUP bar 4 DO desktop
245 --Sets up "picklist" for bar 4
246 DEFINE POPUP menu FROM 7.20 TO 11.47 TITLE "Select Status"
247 MESSAGE "Scroll or press highlighted letter to select a Pano Status"
248 COLOR W/BG W/BG B/BG B/BG B/BG W/M W/R GR/W
249 DEFINE BAR 1 OF PANO PROMPT "VGRN - accepted/on-file"
250 DEFINE BAR 2 OF PANO PROMPT "VRED - duplicated/forwarded"
251 DEFINE BAR 3 OF PANO PROMPT "VYLM - neither of above"
252 ON SELECTION POPUP menu DO desktop
253 --Sets up "picklist" for bar 4
254 DEFINE POPUP menu FROM 1.17 TO 21.49 TITLE "Select UIC"
255 PROMPT FIELD activity_uic "activity:act.name SCROLL"
256 PROMPT FIELD activity_uic "activity:act.name SCROLL"
257 COLOR W/BG W/BG B/BG B/BG W/M W/R
258 ON SELECTION POPUP menu DO desktop
259 --picklist for curriculum number
260 DEFINE POPUP menu FROM 1.29 TO 21.79 TITLE "Select Curriculum"
261 PROMPT FIELD curricul_curr_num "curricul_curr.name SCROLL"
262 MESSAGE "Scroll to locate and select a valid Curriculum"
263 COLOR W/BG W/BG B/BG B/BG W/M W/R
264 ON SELECTION POPUP menu DO desktop
265 RETURN
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

1 * *****
2 *
3 * Procedure file: C:\PTMS\NPS_INTR.PRG
4 *
5 * Proc & Fnct: DELAY
6 *
7 * Set by: NPSOC.PRG
8 *
9 * Call: DELAY
10 * (procedure in NPS_INTR.PRG)
11 * PASSTRAN
12 * GETKEY
13 *
14 * Uses: NPS.MISC.DBF
15 *
16 * Documented 01/28/92 at 09:00 FoxDoc version 2.10
17 * Program: NPS_INTR.PRG (Introductory screen, password test,
18 * data/time check module)
19 * PRIVATE COL.col.cntr,counter,password,i
20 * oldcolor,keycode,choice,returnkey,colget,colget.esax,i
21 * n ---Center the Intro heading
22 * hds = "NPSOC PATIENT TRACKING AND RECALL SYSTEM (PTARS)"
23 * COL = (80 - LEN(hds)) / 2
24 * i = .175
25 * counter = 0
26 * DO WHILE counter < 7
27 * DO CASE
28 * CASE counter = 0
29 * SET COLOR TO BG/N
30 * CASE counter = 1
31 * SET COLOR TO G/N
32 * CASE counter = 2
33 * SET COLOR TO RB/N
34 * CASE counter = 3
35 * SET COLOR TO R/N
36 * CASE counter = 4
37 * SET COLOR TO B/N
38 * CASE counter = 5
39 * SET COLOR TO W/N
40 * CASE counter = 6
41 * SET COLOR TO GR/N
42 * ENDCASE
43 * 0 5.15 SAY -
44 * 0 6.15 SAY -
45 * 0 7.15 SAY -
46 * 0 8.15 SAY -
47 * 0 9.15 SAY -
48 * 0 10.15 SAY -
49 * 0 11.15 SAY -
50 * 0 12.15 SAY -
51 * 0 13.15 SAY -
52 * 0 14.15 SAY -
53 * 0 15.15 SAY -
54 * DO delay WITH x
55 * counter = counter + 1
56 * ENDDO
57 * 0 17.COL-1 SAY "NPSOC Patient Tracking and Recall System (PTARS)"
58 * COLOR GR/N
59 * 0 18.33 SAY "Version 1.0" COLOR GR/N
60 * WAIT -- TIMEOUT 2
61 * CLEAR
62 * 0 2.COL SAY hds COLOR GR/N
63 * 0 3.33 SAY "Access Screen" COLOR GR/N
64 * 0 1.0 TO 8.79 DOUBLE COLOR G/N
65 * 0 4.0 SAY -
66 * 0 4.40 SAY -
67 * SET COLOR TO W/N,N/W
68 * n ---Begin Password validation
69 * n ---Check for existence of PASSWDR.DBF
70 * IF .NOT. FILE("NPS.MISC.DBF")
71 * 77 CHR(7)
72 * WAIT "Password file not found. Program quitting..." WINDOW TIMEOUT 3
73 * SET COLOR TO W/N,N/W
74 * CLEAR
75 * 0 0.0 SAY "NPS.MISC.DBF not found. Check User's Manual." COLOR W/N
76 * CLOSE ALL
77 * QUIT
78 * ENDOF
79 * n ---Using NPS.MISC.DBF
80 * USE nps_disc IN EX
81 * SELECT E
82 * 0 11.35 SAY -
83 * 0 12.35 SAY -
84 * 0 13.35 SAY -
85 * 0 14.35 SAY -
86 * n ---Password routines
87 * counter = 0
88 * DO WHILE .T.

```

```

89 * password = SPACE(4)
90 * 0 6.25 SAY "Please type the password."
91 * 0 6.52 SAY - "COLOR N/W"
92 * 77 CHR(7)
93 * colget = 59
94 * colsay = 51
95 * esax = SPACE(1)
96 * password = ""
97 * FOR i = 1 TO 6
98 * 0 6.colget + 1 GET esax COLOR ,i
99 * READ
100 * password = password+esax
101 * 0 6.colsay + 1 SAY "-"
102 * CLEAR GETS
103 * ENDFOR
104 * IF UPPER(password) = passtran(password)
105 * EXIT
106 * ELSE
107 * 77 CHR(7)
108 * counter = counter + 1
109 * DO CASE
110 * CASE counter < 3
111 * WAIT "Invalid Password on attempt "TRIM(STR(counter))";
112 * Try again ..." WINDOW TIMEOUT 1
113 * CASE counter = 3
114 * WAIT "Invalid Password on attempt "TRIM(STR(counter))";
115 * Quitting program ..." WINDOW TIMEOUT 1
116 * SET COLOR TO W/N
117 * CLEAR
118 * 0 0.0 SAY "Password failure entering PTARS. User rejected."
119 * COLOR W/N
120 * QUIT
121 * ENDCASE
122 * LOOP
123 * ENDOF
124 * ENDDO
125 * USE
126 * n ---Check sysdate date
127 * SET CURSOR ON
128 * 0 6.2 CLEAR TO 6.77
129 * 77 CHR(7)
130 * SET COLOR TO W/N
131 * 0 6.11 SAY "The system date is"
132 * 0 6.30 SAY DATE() COLOR W/N
133 * 0 6.41 SAY "Is this date correct? (y/n)"
134 * choice = "n"
135 * DO gattay WITH choice, "Y"
136 * IF choice = "N"
137 * 0 11.0 CLEAR TO 20.79
138 * 0 11.0 SAY --
139 * RUN DATE
140 * 0 11.0 CLEAR TO 23.79
141 * ENDOF
142 * 0 6.2 CLEAR TO 6.77
143 * 77 CHR(7)
144 * 0 6.11 SAY "The system time is"
145 * 0 6.30 SAY TIME() COLOR W/N
146 * 0 6.41 SAY "Is this time correct? (y/n)"
147 * choice = "n"
148 * DO gattay WITH choice, "Y"
149 * IF choice = "N"
150 * 0 11.0 SAY --
151 * RUN TIME
152 * 0 11.0 CLEAR TO 23.79
153 * ENDOF
154 * SET CURSOR OFF
155 * SET COLOR TO W/N,N/W
156 * 0 1.0 CLEAR TO 10.79
157 * 0 0.40 SAY DATE() COLOR N/W
158 * 0 0.48 SAY - "COLOR N/W"
159 * SET CLOCK ON
160 * RETURN
161 * *****
162 * 1
163 * Procedure: DELAY
164 * 1
165 * Called by: NPS_INTR.PRG
166 * 1
167 * *****
168 * PROCEDURE delay
169 * PRIVATE y
170 * 1
171 * z = SECONDS() + y
172 * DO WHILE SECONDS() < z
173 * ENDDO
174 * RETURN
175 * 1, EOF, NPS_INTR.PRG

```



```

1  * *****
2  * Procedures files: C:\PTARS\NPS_OPEN.PRG
3  *
4  *
5  * Procs & Fncs: WARN_DBF
6  *      SPACE_CHECK
7  *      UPDATE
8  *
9  *
10 * Set bvi, NPSDC.PRG
11 *      PACK_POP
12 *      REST_POP
13 *
14 * Calls: WARN_DBF
15 *      SPACE_CHECK
16 *      UPDATE
17 *      NPS_AREA
18 *
19 * Usses: MEMBERS.DBF
20 *      ACTIVITY.DBF
21 *      CURRICUL.DBF
22 *      DIRECTOR.DBF
23 *      NPS_MISC.DBF
24 *
25 * Indexes: SSN
26 *      NAME
27 *      CLASS
28 *      CLASH
29 *      UICS
30 *      UICN
31 *      CURR_MANS
32 *      PAMOS
33 *      PANON
34 *      ACT_UIC
35 *      CURR_NAME
36 *
37 * CDS files: MEMBERS.CDX
38 *      ACTIVITY.CDX
39 *      CURRICUL.CDX
40 *
41 * Documented 01/28/92 at 09:00 FoxDoc version 2.10
42 * *****
43 * *****
44 * PRIVATE ROM,orig,buf,missing_dbf
45 * SET COLOR TO W/N/M/W
46 * STORE 0 TO choice_eor
47 * psw = CHR(13)
48 * returnkey = CHR(13)
49 * deircard = CHR(7)
50 * Initialize global variables.
51 * STORE 0 TO strc,oldrechu,rechum,menuchoice
52 * STORE 0 TO choice_eor
53 * procmov = 2
54 * *****
55 * Check if all database files are missing
56 * IF ( ( FILE("MEMBERS.DBF") ) AND ( ( FILE("ACTIVITY.DBF") ) AND ( ( FILE("CURRICUL.DBF") ) AND ( ( FILE("DIRECTOR.DBF") ) ) ) ) ) )
57 * missing_dbf = "ALL databases"
58 * no_dbf = .T.
59 * DO warn_dbf WITH missing_dbf
60 * RETURN
61 * ENDIF
62 *
63 * ---Open members database file.
64 * IF NOT FILE("MEMBERS.DBF")
65 * missing_dbf = "MEMBERS.DBF"
66 * no_dbf = .T.
67 * DO warn_dbf WITH missing_dbf
68 * RETURN
69 * ENDIF
70 *
71 * USE members
72 *
73 * IF ( ( FILE("members.cdx") ) OR ( ( FILE("activity.cdx") ) OR ( ( FILE("curricul.cdx") ) ) ) )
74 * IF 0 19.0 CLEAR TO 23.79
75 * IF 0 19.0 SAY "Checking/creating structural index file(s) ...." COLOR W/N
76 * WAIT -- TIMEOUT 1
77 *
78 * ENDIF
79 *
80 * IF NOT FILE("MEMBERS.CDX")
81 * IF 0 24.0 SAY "Working ...." COLOR W/N
82 * INDEX ON SSN TAG SSN ADDITIVE
83 * INDEX ON SSN TAG SSN ADDITIVE
84 * INDEX ON last_name-first_name TAG class ADDITIVE
85 * INDEX ON STR(class)last_name-first_name TAG class ADDITIVE
86 * INDEX ON uicssn TAG uics ADDITIVE
87 * INDEX ON uiclast_name-first_name TAG uicn ADDITIVE
88

```

```

89 * INDEX ON curr_num TAG curr_num ADDITIVE
90 * INDEX ON panon TAG panon ADDITIVE
91 * INDEX ON panonlast_name-first_name TAG panon ADDITIVE
92 * WAIT -- TIMEOUT 1
93 * ENDIF
94 *
95 * ---Do initial startup checks
96 * IF ( ( open_checks )
97 * ---Check if sufficient diskspaces available
98 * DO spaces_check
99 * ---Change members class to 4 if full 12 mos since last_12
100 * DO UPDATE
101 * ENDIF
102 * ---Open activity database file.
103 * IF NOT FILE("ACTIVITY.DBF")
104 * missing_dbf = "ACTIVITY.DBF"
105 * no_dbf = .T.
106 * DO warn_dbf WITH missing_dbf
107 * RETURN
108 * ENDIF
109 *
110 * USE activity
111 *
112 * ---Open index file.
113 * IF NOT FILE("ACTIVITY.CDX")
114 * IF 0 20.0 SAY "Creating structural index 'ACTIVITY.CDX' ..."
115 * INDEX ON uic TAG act_uic ADDITIVE
116 * WAIT -- TIMEOUT 1
117 * ENDIF
118 *
119 * ---Open curriculum database file.
120 * IF NOT FILE("CURRICUL.DBF")
121 * missing_dbf = "CURRICUL.DBF"
122 * no_dbf = .T.
123 * DO warn_dbf WITH missing_dbf
124 * RETURN
125 * ENDIF
126 *
127 * SELECT 3
128 * USE curricul
129 *
130 * ---open index file.
131 * IF NOT FILE("CURRICUL.CDX")
132 * IF 0 20.0 SAY "Creating structural index 'CURRICUL.CDX' ..."
133 * INDEX ON curr_num TAG curr_num ADDITIVE
134 * WAIT -- TIMEOUT 1
135 * ENDIF
136 *
137 * ---open director database file.
138 * IF NOT FILE("DIRECTOR.DBF")
139 * missing_dbf = "DIRECTOR.DBF"
140 * no_dbf = .T.
141 * DO warn_dbf WITH missing_dbf
142 * RETURN
143 * ENDIF
144 *
145 * USE DIRECTOR
146 *
147 * ---Open members database file.
148 * IF NOT FILE("MEMBERS.DBF")
149 * missing_dbf = "MEMBERS.DBF"
150 * no_dbf = .T.
151 * DO warn_dbf WITH missing_dbf
152 * RETURN
153 * ENDIF
154 *
155 * ---Set relation(i).
156 * IF NOT FILE("RELATE1.VUE") OR. norelatel = .T.
157 * SET RELATION TO uic INTO members ADDITIVE
158 * SELECT 2
159 * SET RELATION TO curr_num INTO members ADDITIVE
160 * SELECT 1
161 * ---Save environment to view file i.
162 * CREATE VIEW relat1 FROM ENVIRONMENT
163 * norelatel = .F.
164 * ENDIF
165 *
166 * CLOSE DATABASES
167 * IF NOT FILE("RELATE2.VUE") OR. norelatel2 = .T.
168 * SELECT 1
169 * USE members ORDER 1
170 * SET RELATION IN B ORDER 1
171 * SET RELATION OFF INTO members
172 * SET CURRICUL IN C ORDER 1
173 * SET RELATION OFF INTO members
174 * SET RELATION IN D
175 * USE nos_misc IN E

```

```

176 * SET RELATION TO uic INTO activity ADDITIVE
177 * SET RELATION TO curr_num INTO curricul ADDITIVE
178 * ---Save environment to view file 2
179 * CREATE VIEW relat2 FROM ENVIRONMENT
180 * norelatel2 = .F.
181 * ENDIF
182 *
183 * ---Initialize database variables for current workarea
184 * SET VIEW TO relat1
185 * ---Set up work area definitions
186 * DO nps_area
187 * no_dbf = .F.
188 * ---Small database files are present
189 * RETURN
190 *
191 * *****
192 *
193 * *****
194 *
195 * *****
196 *
197 * *****
198 *
199 * *****
200 *
201 * *****
202 *
203 * *****
204 *
205 * *****
206 *
207 * *****
208 *
209 * *****
210 *
211 * *****
212 *
213 * *****
214 *
215 * *****
216 *
217 * *****
218 *
219 * *****
220 *
221 * *****
222 *
223 * *****
224 *
225 * *****
226 *
227 * *****
228 *
229 * *****
230 *
231 * *****
232 *
233 * *****
234 *
235 * *****
236 *
237 * *****
238 *
239 * *****
240 *
241 * *****
242 *
243 * *****
244 *
245 * *****
246 *
247 * *****
248 *
249 * *****
250 *
251 * *****
252 *
253 * *****

```

```

1 *****
2
3 Procedure file, C:\PTARS\NPS_APPE.PRG
4
5 Proc & Fncs, NPS_SAV1
6 NPS_SAV2
7 NPS_SAV3
8 NPS_CHECK
9
10 Set br, NPSLOC.PRG
11
12 Calls, NPS_FORTHAI
13 NPS_FORTHB
14 NPS_FORTHC
15 NPS_FORTH
16 NPS_SAV1
17 NPS_SAV2
18 NPS_SAV3
19 NPS_STOR
20 NPS_STOR
21 NPS_KEYS
22 NPS_KEYS
23 NPS_GETS
24 NPS_GETS
25 NPS_CHECK
26 NPS_CHECK
27 NPS_REPL
28
29 *****
30 Documented 01/28/92 at 09:00 FoxDoc version 2.10
31
32 *****
33 STORE .F. To isblank.isunique.isdeleted.scrnId
34
35 isdeleted = .F.
36
37 NPS = .F.
38
39 *****Set Caplock off
40
41 NPS = .F.
42
43 *****Select Input form
44
45 NPS = .F.
46
47 *****Set Caplock off
48
49 NPS = .F.
50
51 *****Set Caplock off
52
53 NPS = .F.
54
55 *****Set Caplock off
56
57 NPS = .F.
58
59 *****Set Caplock off
60
61 NPS = .F.
62
63 *****Set Caplock off
64
65 NPS = .F.
66
67 *****Set Caplock off
68
69 NPS = .F.
70
71 *****Set Caplock off
72
73 NPS = .F.
74
75 *****Set Caplock off
76
77 NPS = .F.
78
79 *****Set Caplock off
80
81 NPS = .F.
82
83 *****Set Caplock off
84
85 NPS = .F.
86
87 *****Set Caplock off
88
89 NPS = .F.
90
91 *****Set Caplock off
92
93 NPS = .F.
94
95 *****Set Caplock off
96
97 NPS = .F.
98
99 *****Set Caplock off
100
101 NPS = .F.
102
103 *****Set Caplock off
104
105 NPS = .F.
106
107 *****Set Caplock off
108
109 NPS = .F.
110
111 *****Set Caplock off
112
113 NPS = .F.
114
115 *****Set Caplock off
116
117 NPS = .F.
118
119 *****Set Caplock off
120
121 NPS = .F.
122
123 *****Set Caplock off
124
125 NPS = .F.
126
127 *****Set Caplock off
128
129 NPS = .F.
130
131 *****Set Caplock off
132
133 NPS = .F.
134
135 *****Set Caplock off
136
137 NPS = .F.
138
139 *****Set Caplock off
140
141 NPS = .F.
142
143 *****Set Caplock off
144
145 NPS = .F.
146
147 *****Set Caplock off
148
149 NPS = .F.
150
151 *****Set Caplock off
152
153 NPS = .F.
154
155 *****Set Caplock off
156
157 NPS = .F.
158
159 *****Set Caplock off
160
161 NPS = .F.
162
163 *****Set Caplock off
164
165 NPS = .F.
166
167 *****Set Caplock off
168
169 NPS = .F.
170
171 *****Set Caplock off
172
173 NPS = .F.
174
175 *****Set Caplock off
176
177 NPS = .F.
178
179 *****Set Caplock off
180
181 NPS = .F.
182
183 *****Set Caplock off
184
185 NPS = .F.
186
187 *****Set Caplock off
188
189 NPS = .F.
190
191 *****Set Caplock off
192
193 NPS = .F.
194
195 *****Set Caplock off
196
197 NPS = .F.
198
199 *****Set Caplock off
200
201 NPS = .F.
202
203 *****Set Caplock off
204
205 NPS = .F.
206
207 *****Set Caplock off
208
209 NPS = .F.
210
211 *****Set Caplock off
212
213 NPS = .F.
214
215 *****Set Caplock off
216
217 NPS = .F.
218
219 *****Set Caplock off
220
221 NPS = .F.
222
223 *****Set Caplock off
224
225 NPS = .F.
226
227 *****Set Caplock off
228
229 NPS = .F.
230
231 *****Set Caplock off
232
233 NPS = .F.
234
235 *****Set Caplock off
236
237 NPS = .F.
238
239 *****Set Caplock off
240
241 NPS = .F.
242
243 *****Set Caplock off
244
245 NPS = .F.
246
247 *****Set Caplock off
248
249 NPS = .F.
250
251 *****Set Caplock off
252
253 NPS = .F.
254
255 *****Set Caplock off
256
257 NPS = .F.
258
259 *****Set Caplock off
260
261 NPS = .F.
262
263 *****Set Caplock off
264
265 NPS = .F.
266
267 *****Set Caplock off
268
269 NPS = .F.
270
271 *****Set Caplock off
272
273 NPS = .F.
274
275 *****Set Caplock off
276
277 NPS = .F.
278
279 *****Set Caplock off
280
281 NPS = .F.
282
283 *****Set Caplock off
284
285 NPS = .F.
286
287 *****Set Caplock off
288
289 NPS = .F.
290
291 *****Set Caplock off
292
293 NPS = .F.
294
295 *****Set Caplock off
296
297 NPS = .F.
298
299 *****Set Caplock off
300
301 NPS = .F.
302
303 *****Set Caplock off
304
305 NPS = .F.
306
307 *****Set Caplock off
308
309 NPS = .F.
310
311 *****Set Caplock off
312
313 NPS = .F.
314
315 *****Set Caplock off
316
317 NPS = .F.
318
319 *****Set Caplock off
320
321 NPS = .F.
322
323 *****Set Caplock off
324
325 NPS = .F.
326
327 *****Set Caplock off
328
329 NPS = .F.
330
331 *****Set Caplock off
332
333 NPS = .F.
334
335 *****Set Caplock off
336
337 NPS = .F.
338
339 *****Set Caplock off
340
341 NPS = .F.
342
343 *****Set Caplock off
344
345 NPS = .F.
346
347 *****Set Caplock off
348
349 NPS = .F.
350
351 *****Set Caplock off
352
353 NPS = .F.
354
355 *****Set Caplock off
356
357 NPS = .F.
358
359 *****Set Caplock off
360
361 NPS = .F.
362
363 *****Set Caplock off
364
365 NPS = .F.
366
367 *****Set Caplock off
368
369 NPS = .F.
370
371 *****Set Caplock off
372
373 NPS = .F.
374
375 *****Set Caplock off
376
377 NPS = .F.
378
379 *****Set Caplock off
380
381 NPS = .F.
382
383 *****Set Caplock off
384
385 NPS = .F.
386
387 *****Set Caplock off
388
389 NPS = .F.
390
391 *****Set Caplock off
392
393 NPS = .F.
394
395 *****Set Caplock off
396
397 NPS = .F.
398
399 *****Set Caplock off
400
401 NPS = .F.
402
403 *****Set Caplock off
404
405 NPS = .F.
406
407 *****Set Caplock off
408
409 NPS = .F.
410
411 *****Set Caplock off
412
413 NPS = .F.
414
415 *****Set Caplock off
416
417 NPS = .F.
418
419 *****Set Caplock off
420
421 NPS = .F.
422
423 *****Set Caplock off
424
425 NPS = .F.
426
427 *****Set Caplock off
428
429 NPS = .F.
430
431 *****Set Caplock off
432
433 NPS = .F.
434
435 *****Set Caplock off
436
437 NPS = .F.
438
439 *****Set Caplock off
440
441 NPS = .F.
442
443 *****Set Caplock off
444
445 NPS = .F.
446
447 *****Set Caplock off
448
449 NPS = .F.
450
451 *****Set Caplock off
452
453 NPS = .F.
454
455 *****Set Caplock off
456
457 NPS = .F.
458
459 *****Set Caplock off
460
461 NPS = .F.
462
463 *****Set Caplock off
464
465 NPS = .F.
466
467 *****Set Caplock off
468
469 NPS = .F.
470
471 *****Set Caplock off
472
473 NPS = .F.
474
475 *****Set Caplock off
476
477 NPS = .F.
478
479 *****Set Caplock off
480
481 NPS = .F.
482
483 *****Set Caplock off
484
485 NPS = .F.
486
487 *****Set Caplock off
488
489 NPS = .F.
490
491 *****Set Caplock off
492
493 NPS = .F.
494
495 *****Set Caplock off
496
497 NPS = .F.
498
499 *****Set Caplock off
500
501 NPS = .F.
502
503 *****Set Caplock off
504
505 NPS = .F.
506
507 *****Set Caplock off
508
509 NPS = .F.
510
511 *****Set Caplock off
512
513 NPS = .F.
514
515 *****Set Caplock off
516
517 NPS = .F.
518
519 *****Set Caplock off
520
521 NPS = .F.
522
523 *****Set Caplock off
524
525 NPS = .F.
526
527 *****Set Caplock off
528
529 NPS = .F.
530
531 *****Set Caplock off
532
533 NPS = .F.
534
535 *****Set Caplock off
536
537 NPS = .F.
538
539 *****Set Caplock off
540
541 NPS = .F.
542
543 *****Set Caplock off
544
545 NPS = .F.
546
547 *****Set Caplock off
548
549 NPS = .F.
550
551 *****Set Caplock off
552
553 NPS = .F.
554
555 *****Set Caplock off
556
557 NPS = .F.
558
559 *****Set Caplock off
560
561 NPS = .F.
562
563 *****Set Caplock off
564
565 NPS = .F.
566
567 *****Set Caplock off
568
569 NPS = .F.
570
571 *****Set Caplock off
572
573 NPS = .F.
574
575 *****Set Caplock off
576
577 NPS = .F.
578
579 *****Set Caplock off
580
581 NPS = .F.
582
583 *****Set Caplock off
584
585 NPS = .F.
586
587 *****Set Caplock off
588
589 NPS = .F.
590
591 *****Set Caplock off
592
593 NPS = .F.
594
595 *****Set Caplock off
596
597 NPS = .F.
598
599 *****Set Caplock off
600
601 NPS = .F.
602
603 *****Set Caplock off
604
605 NPS = .F.
606
607 *****Set Caplock off
608
609 NPS = .F.
610
611 *****Set Caplock off
612
613 NPS = .F.
614
615 *****Set Caplock off
616
617 NPS = .F.
618
619 *****Set Caplock off
620
621 NPS = .F.
622
623 *****Set Caplock off
624
625 NPS = .F.
626
627 *****Set Caplock off
628
629 NPS = .F.
630
631 *****Set Caplock off
632
633 NPS = .F.
634
635 *****Set Caplock off
636
637 NPS = .F.
638
639 *****Set Caplock off
640
641 NPS = .F.
642
643 *****Set Caplock off
644
645 NPS = .F.
646
647 *****Set Caplock off
648
649 NPS = .F.
650
651 *****Set Caplock off
652
653 NPS = .F.
654
655 *****Set Caplock off
656
657 NPS = .F.
658
659 *****Set Caplock off
660
661 NPS = .F.
662
663 *****Set Caplock off
664
665 NPS = .F.
666
667 *****Set Caplock off
668
669 NPS = .F.
670
671 *****Set Caplock off
672
673 NPS = .F.
674
675 *****Set Caplock off
676
677 NPS = .F.
678
679 *****Set Caplock off
680
681 NPS = .F.
682
683 *****Set Caplock off
684
685 NPS = .F.
686
687 *****Set Caplock off
688
689 NPS = .F.
690
691 *****Set Caplock off
692
693 NPS = .F.
694
695 *****Set Caplock off
696
697 NPS = .F.
698
699 *****Set Caplock off
700
701 NPS = .F.
702
703 *****Set Caplock off
704
705 NPS = .F.
706
707 *****Set Caplock off
708
709 NPS = .F.
710
711 *****Set Caplock off
712
713 NPS = .F.
714
715 *****Set Caplock off
716
717 NPS = .F.
718
719 *****Set Caplock off
720
721 NPS = .F.
722
723 *****Set Caplock off
724
725 NPS = .F.
726
727 *****Set Caplock off
728
729 NPS = .F.
730
731 *****Set Caplock off
732
733 NPS = .F.
734
735 *****Set Caplock off
736
737 NPS = .F.
738
739 *****Set Caplock off
740
741 NPS = .F.
742
743 *****Set Caplock off
744
745 NPS = .F.
746
747 *****Set Caplock off
748
749 NPS = .F.
750
751 *****Set Caplock off
752
753 NPS = .F.
754
755 *****Set Caplock off
756
757 NPS = .F.
758
759 *****Set Caplock off
760
761 NPS = .F.
762
763 *****Set Caplock off
764
765 NPS = .F.
766
767 *****Set Caplock off
768
769 NPS = .F.
770
771 *****Set Caplock off
772
773 NPS = .F.
774
775 *****Set Caplock off
776
777 NPS = .F.
778
779 *****Set Caplock off
780
781 NPS = .F.
782
783 *****Set Caplock off
784
785 NPS = .F.
786
787 *****Set Caplock off
788
789 NPS = .F.
790
791 *****Set Caplock off
792
793 NPS = .F.
794
795 *****Set Caplock off
796
797 NPS = .F.
798
799 *****Set Caplock off
800
801 NPS = .F.
802
803 *****Set Caplock off
804
805 NPS = .F.
806
807 *****Set Caplock off
808
809 NPS = .F.
810
811 *****Set Caplock off
812
813 NPS = .F.
814
815 *****Set Caplock off
816
817 NPS = .F.
818
819 *****Set Caplock off
820
821 NPS = .F.
822
823 *****Set Caplock off
824
825 NPS = .F.
826
827 *****Set Caplock off
828
829 NPS = .F.
830
831 *****Set Caplock off
832
833 NPS = .F.
834
835 *****Set Caplock off
836
837 NPS = .F.
838
839 *****Set Caplock off
840
841 NPS = .F.
842
843 *****Set Caplock off
844
845 NPS = .F.
846
847 *****Set Caplock off
848
849 NPS = .F.
850
851 *****Set Caplock off
852
853 NPS = .F.
854
855 *****Set Caplock off
856
857 NPS = .F.
858
859 *****Set Caplock off
860
861 NPS = .F.
862
863 *****Set Caplock off
864
865 NPS = .F.
866
867 *****Set Caplock off
868
869 NPS = .F.
870
871 *****Set Caplock off
872
873 NPS = .F.
874
875 *****Set Caplock off
876
877 NPS = .F.
878
879 *****Set Caplock off
880
881 NPS = .F.
882
883 *****Set Caplock off
884
885 NPS = .F.
886
887 *****Set Caplock off
888
889 NPS = .F.
890
891 *****Set Caplock off
892
893 NPS = .F.
894
895 *****Set Caplock off
896
897 NPS = .F.
898
899 *****Set Caplock off
900
901 NPS = .F.
902
903 *****Set Caplock off
904
905 NPS = .F.
906
907 *****Set Caplock off
908
909 NPS = .F.
910
911 *****Set Caplock off
912
913 NPS = .F.
914
915 *****Set Caplock off
916
917 NPS = .F.
918
919 *****Set Caplock off
920
921 NPS = .F.
922
923 *****Set Caplock off
924
925 NPS = .F.
926
927 *****Set Caplock off
928
929 NPS = .F.
930
931 *****Set Caplock off
932
933 NPS = .F.
934
935 *****Set Caplock off
936
937 NPS = .F.
938
939 *****Set Caplock off
940
941 NPS = .F.
942
943 *****Set Caplock off
944
945 NPS = .F.
946
947 *****Set Caplock off
948
949 NPS = .F.
950
951 *****Set Caplock off
952
953 NPS = .F.
954
955 *****Set Caplock off
956
957 NPS = .F.
958
959 *****Set Caplock off
960
961 NPS = .F.
962
963 *****Set Caplock off
964
965 NPS = .F.
966
967 *****Set Caplock off
968
969 NPS = .F.
970
971 *****Set Caplock off
972
973 NPS = .F.
974
975 *****Set Caplock off
976
977 NPS = .F.
978
979 *****Set Caplock off
980
981 NPS = .F.
982
983 *****Set Caplock off
984
985 NPS = .F.
986
987 *****Set Caplock off
988
989 NPS = .F.
990
991 *****Set Caplock off
992
993 NPS = .F.
994
995 *****Set Caplock off
996
997 NPS = .F.
998
999 *****Set Caplock off
1000
1001 NPS = .F.

```

```

89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```



```

262 WINDOW TIMEOUT 2
263 isolated = .T.
264 ENDIF
265 CASE @bfarea = "3"
266 IF (LEN(ALLTRIM(scurr_name)) < 3) OR (EMPTY(ALLTRIM(scurr_name)))
267 77 CHR(7)
268 WAIT - Incomplete or missing data. Record not saved.
269 WINDOW TIMEOUT 2
270 isolated = .T.
271 ENDIF
272 ENDCASE
273 RETURN
274 M, EOP, NPS_APPE.PRG

```

```

*) at .12
177 IF (wlc (> "31405")
178     REPLACE curr_numb WITH ""
179 ENDIF
180 ELSE
181     WAIT "Record edit aborted." WINDOW TIMEOUT 1
182 ENDIF
183 ENDOFSE
184 ENDCASE
185 ENDDO
186 RETURN
187 /* End of procedure update */
188 **
189 ** Procedure, UPDATEDEC
190 **
191 ** Called by: MPS_EDIT.PRG
192 **
193 **
194 **
195 PROCEDURE updatedec
196 **
197 **
198 **
199 **
200 ENDOF
201 RETURN
202 **
203 **
204 ** Procedure, DIRECTOR
205 **
206 **
207 ** Called by: MPS_EDIT.PRG
208 **
209 **
210 ** Calls: HLP
211 **
212 **
213 **
214 **
215 **
216 **
217 **
218 **
219 **
220 **
221 **
222 **
223 **
224 **
225 **
226 **
227 **
228 **
229 **
230 **
231 **
232 **
233 **
234 **
235 **
236 **
237 **
238 **
239 **
240 **
241 **
242 **
243 **
244 **
245 **
246 **
247 **
248 **
249 **
250 **
251 **
252 **
253 **
254 **
255 **
256 **
257 **
258 **
259 **
260 **
261 **
262 **
263 **
264 **
265 **
266 **
267 **
268 **
269 **
270 **
271 **
272 **
273 **
274 **
275 **
276 **
277 **
278 **
279 **
280 **
281 **
282 **
283 **
284 **
285 **
286 **
287 **
288 **
289 **
290 **
291 **
292 **
293 **
294 **
295 **
296 **
297 **
298 **
299 **
300 **
301 **
302 **
303 **
304 **
305 **
306 **
307 **
308 **
309 **
310 **
311 **
312 **
313 **
314 **
315 **
316 **
317 **
318 **
319 **
320 **
321 **
322 **
323 **
324 **
325 **
326 **
327 **
328 **
329 **
330 **
331 **
332 **
333 **
334 **
335 **
336 **
337 **
338 **
339 **
340 **
341 **
342 **
343 **
344 **
345 **
346 **
347 **
348 **
349 **
350 **
351 **
352 **
353 **
354 **
355 **
356 **
357 **
358 **
359 **
360 **
361 **
362 **
363 **
364 **
365 **
366 **
367 **
368 **
369 **
370 **
371 **
372 **
373 **
374 **
375 **
376 **
377 **
378 **
379 **
380 **
381 **
382 **
383 **
384 **
385 **
386 **
387 **
388 **
389 **
390 **
391 **
392 **
393 **
394 **
395 **
396 **
397 **
398 **
399 **
400 **
401 **
402 **
403 **
404 **
405 **
406 **
407 **
408 **
409 **
410 **
411 **
412 **
413 **
414 **
415 **
416 **
417 **
418 **
419 **
420 **
421 **
422 **
423 **
424 **
425 **
426 **
427 **
428 **
429 **
430 **
431 **
432 **
433 **
434 **
435 **
436 **
437 **
438 **
439 **
440 **
441 **
442 **
443 **
444 **
445 **
446 **
447 **
448 **
449 **
450 **
451 **
452 **
453 **
454 **
455 **
456 **
457 **
458 **
459 **
460 **
461 **
462 **
463 **
464 **
465 **
466 **
467 **
468 **
469 **
470 **
471 **
472 **
473 **
474 **
475 **
476 **
477 **
478 **
479 **
480 **
481 **
482 **
483 **
484 **
485 **
486 **
487 **
488 **
489 **
490 **
491 **
492 **
493 **
494 **
495 **
496 **
497 **
498 **
499 **
500 **
501 **
502 **
503 **
504 **
505 **
506 **
507 **
508 **
509 **
510 **
511 **
512 **
513 **
514 **
515 **
516 **
517 **
518 **
519 **
520 **
521 **
522 **
523 **
524 **
525 **
526 **
527 **
528 **
529 **
530 **
531 **
532 **
533 **
534 **
535 **
536 **
537 **
538 **
539 **
540 **
541 **
542 **
543 **
544 **
545 **
546 **
547 **
548 **
549 **
550 **
551 **
552 **
553 **
554 **
555 **
556 **
557 **
558 **
559 **
560 **
561 **
562 **
563 **
564 **
565 **
566 **
567 **
568 **
569 **
570 **
571 **
572 **
573 **
574 **
575 **
576 **
577 **
578 **
579 **
580 **
581 **
582 **
583 **
584 **
585 **
586 **
587 **
588 **
589 **
590 **
591 **
592 **
593 **
594 **
595 **
596 **
597 **
598 **
599 **
600 **
601 **
602 **
603 **
604 **
605 **
606 **
607 **
608 **
609 **
610 **
611 **
612 **
613 **
614 **
615 **
616 **
617 **
618 **
619 **
620 **
621 **
622 **
623 **
624 **
625 **
626 **
627 **
628 **
629 **
630 **
631 **
632 **
633 **
634 **
635 **
636 **
637 **
638 **
639 **
640 **
641 **
642 **
643 **
644 **
645 **
646 **
647 **
648 **
649 **
650 **
651 **
652 **
653 **
654 **
655 **
656 **
657 **
658 **
659 **
660 **
661 **
662 **
663 **
664 **
665 **
666 **
667 **
668 **
669 **
670 **
671 **
672 **
673 **
674 **
675 **
676 **
677 **
678 **
679 **
680 **
681 **
682 **
683 **
684 **
685 **
686 **
687 **
688 **
689 **
690 **
691 **
692 **
693 **
694 **
695 **
696 **
697 **
698 **
699 **
700 **
701 **
702 **
703 **
704 **
705 **
706 **
707 **
708 **
709 **
710 **
711 **
712 **
713 **
714 **
715 **
716 **
717 **
718 **
719 **
720 **
721 **
722 **
723 **
724 **
725 **
726 **
727 **
728 **
729 **
730 **
731 **
732 **
733 **
734 **
735 **
736 **
737 **
738 **
739 **
740 **
741 **
742 **
743 **
744 **
745 **
746 **
747 **
748 **
749 **
750 **
751 **
752 **
753 **
754 **
755 **
756 **
757 **
758 **
759 **
760 **
761 **
762 **
763 **
764 **
765 **
766 **
767 **
768 **
769 **
770 **
771 **
772 **
773 **
774 **
775 **
776 **
777 **
778 **
779 **
780 **
781 **
782 **
783 **
784 **
785 **
786 **
787 **
788 **
789 **
790 **
791 **
792 **
793 **
794 **
795 **
796 **
797 **
798 **
799 **
800 **
801 **
802 **
803 **
804 **
805 **
806 **
807 **
808 **
809 **
810 **
811 **
812 **
813 **
814 **
815 **
816 **
817 **
818 **
819 **
820 **
821 **
822 **
823 **
824 **
825 **
826 **
827 **
828 **
829 **
830 **
831 **
832 **
833 **
834 **
835 **
836 **
837 **
838 **
839 **
840 **
841 **
842 **
843 **
844 **
845 **
846 **
847 **
848 **
849 **
850 **
851 **
852 **
853 **
854 **
855 **
856 **
857 **
858 **
859 **
860 **
861 **
862 **
863 **
864 **
865 **
866 **
867 **
868 **
869 **
870 **
871 **
872 **
873 **
874 **
875 **
876 **
877 **
878 **
879 **
880 **
881 **
882 **
883 **
884 **
885 **
886 **
887 **
888 **
889 **
890 **
891 **
892 **
893 **
894 **
895 **
896 **
897 **
898 **
899 **
900 **
901 **
902 **
903 **
904 **
905 **
906 **
907 **
908 **
909 **
910 **
911 **
912 **
913 **
914 **
915 **
916 **
917 **
918 **
919 **
920 **
921 **
922 **
923 **
924 **
925 **
926 **
927 **
928 **
929 **
930 **
931 **
932 **
933 **
934 **
935 **
936 **
937 **
938 **
939 **
940 **
941 **
942 **
943 **
944 **
945 **
946 **
947 **
948 **
949 **
950 **
951 **
952 **
953 **
954 **
955 **
956 **
957 **
958 **
959 **
960 **
961 **
962 **
963 **
964 **
965 **
966 **
967 **
968 **
969 **
970 **
971 **
972 **
973 **
974 **
975 **
976 **
977 **
978 **
979
```

```

90  ENDIF
91  CASE editchoice = "G"
92  M ---Goto a record
93  DO goto WITH ROW,rcnum,istrec
94  DO sysrec
95  leadited = .F.
96  CASE editchoice = "N"
97  oldrcnum = RECH01)
98  SKIP + 1
99  IF EOF()
100  DO saveof WITH ROW,oldrcnum
101  ELSE
102  DO sysrec
103  leadited = .F.
104  ENDIF
105  CASE editchoice = "P"
106  M ---Previous record.
107  oldrcnum = RECH01)
108  SKIP -1
109  IF BOF()
110  DO saveof WITH ROW,oldrcnum
111  ELSE
112  DO sysrec
113  leadited = .F.
114  ENDIF
115  CASE editchoice = "E"
116  M ---Edit the record.
117  leadited = .T.
118  DO nps.stor
119  oldrcnum = RECH01)
120  M ---Check for duplicate record.
121  DO CASE
122  CASE oldrcnum = "1"
123  keynow = esan
124  CASE oldrcnum = "2"
125  keynow = mulcb
126  CASE oldrcnum = "3"
127  keynow = secur_nmac
128  ENDCASE
129  DO WHILE .T.
130  @ ROW, 0 SAY "EDIT: Press <Esc> to abort"&SPACE(52) COLOR 4//BG
131  SET CURSOR ON
132  M ---Enter key field values.
133  DO nps.keyf WITH exp,1,blank,1,unique
134  IF exp = keynow &&1 original key value
135  EXIT
136  ENDIF
137  IF 1,blank .OR. .NOT. 1,unique
138  EXIT
139  ENDIF
140  M ---Check for duplicate key in master file.
141  SET DELETED OFF
142  SEEK expf
143  IF EOF()
144  M ---No duplicate key found, so leave.
145  SET DELETED ON
146  GO oldrcnum
147  EXIT
148  ELSE
149  M ---Found a duplicate record in the file.
150  DO CASE
151  CASE oldrcnum = "1"
152  77 CHR(7)
153  WAIT "DUPLICATE SSN. Change value to proceed." WINDOW 1
154  TIMEOUT 2
155  CASE oldrcnum = "2"
156  77 CHR(7)
157  WAIT "DUPLICATE UIC. Change value to proceed." WINDOW 1
158  TIMEOUT 2
159  CASE oldrcnum = "3"
160  77 CHR(7)
161  WAIT "DUPLICATE Curriculum. Change value to proceed." ;
162  WINDOW 1
163  TIMEOUT 2
164  ENDCASE
165  GO oldrcnum
166  ENDIF
167  ENDDO
168  IF 1,blank
169  isdeleted = .T.
170  ELSE
171  DO nps.gets
172  ENDIF
173  IF NOT LASTKEY() = 27
174  DO nps.rep1
175  IF oldrcnum = "1"
176  DO updatrec
177  CLEAR recall dates with new

```

NPS EDIT.PRG 1 of 1

```

1  *****
2  1
3  2
4  3 Procedure filler, C:\VPARS\WPS_EDIT.PRG
5
6  Procs & Fncs: DIRECTOR
7  ; UPDATEREC
8
9  Set bvr, WPSDOC.PRG
10
11 Calls:
12 WPS.FORMA2 (procedure in WPS.PROC.PRG)
13 WPS.FOROB (procedure in WPS.PROC.PRG)
14 WPS.FOROB (procedure in WPS.PROC.PRG)
15 DIRECTOR (procedure in WPS.EDIT.PRG)
16 HLP (procedure in WPS.PROC.PRG)
17 SAYREC (procedure in WPS.PROC.PRG)
18 GETKEY (procedure in WPS.PROC.PRG)
19 WPS.NDIS (procedure in WPS.PROC.PRG)
20 WPS.SEEK (procedure in WPS.PROC.PRG)
21 SAYLINE (procedure in WPS.PROC.PRG)
22 DODOTO (procedure in WPS.PROC.PRG)
23 SAYEOF (procedure in WPS.PROC.PRG)
24 WPS.STOR (procedure in WPS.PROC.PRG)
25 WPS.KEYS (procedure in WPS.PROC.PRG)
26 WPS.GETIS (procedure in WPS.PROC.PRG)
27 WPS.REPL (procedure in WPS.PROC.PRG)
28 UPDATEREC (procedure in WPS.EDIT.PRG)
29
30 *****
31 Documented 01/28/92 at 09:00 F:\doc version 2.10
32 *****
33 *****
34 *****
35 *****
36 *****
37 *****
38 *****
39 *****
40 *****
41 *****
42 *****
43 *****
44 *****
45 *****
46 *****
47 *****
48 *****
49 *****
50 *****
51 *****
52 *****
53 *****
54 *****
55 *****
56 *****
57 *****
58 *****
59 *****
60 *****
61 *****
62 *****
63 *****
64 *****
65 *****
66 *****
67 *****
68 *****
69 *****
70 *****
71 *****
72 *****
73 *****
74 *****
75 *****
76 *****
77 *****
78 *****
79 *****
80 *****
81 *****
82 *****
83 *****
84 *****
85 *****
86 *****
87 *****
88 *****

```

```

1  * *****
2  * Procedure file: C:\PTMS\NPS_BROW.PRG
3  *
4  *
5  * Proc & Func: PAINT
6  *
7  *
8  *
9  *
10 *
11 *
12 * Call: HLP
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *
67 *
68 *
69 *
70 *
71 *
72 *
73 *
74 *
75 *
76 *
77 *
78 *
79 *
80 *
81 *
82 *
83 *
84 *
85 *
86 *
87 *
88 *

```

```

89  *
90  *
91  *
92  *
93  *
94  *
95  *
96  *
97  *
98  *
99  *
100 *
101 *
102 *
103 *
104 *
105 *
106 *
107 *
108 *
109 *
110 *
111 *
112 *
113 *
114 *
115 *
116 *
117 *
118 *
119 *
120 *
121 *
122 *
123 *
124 *
125 *
126 *
127 *
128 *
129 *
130 *
131 *
132 *
133 *
134 *
135 *
136 *
137 *
138 *
139 *
140 *
141 *
142 *
143 *
144 *
145 *
146 *
147 *
148 *
149 *
150 *
151 *
152 *
153 *
154 *
155 *
156 *
157 *
158 *
159 *
160 *
161 *
162 *
163 *
164 *
165 *
166 *
167 *
168 *
169 *
170 *
171 *
172 *
173 *
174 *
175 *
176 *

```

```

177 *
178 *
179 *
180 *
181 *
182 *
183 *
184 *
185 *
186 *
187 *
188 *
189 *
190 *
191 *
192 *
193 *
194 *
195 *
196 *
197 *
198 *
199 *
200 *
201 *
202 *
203 *
204 *
205 *
206 *
207 *
208 *
209 *
210 *
211 *
212 *
213 *
214 *
215 *
216 *
217 *
218 *
219 *
220 *
221 *
222 *
223 *
224 *
225 *
226 *
227 *
228 *
229 *
230 *
231 *
232 *
233 *
234 *
235 *
236 *
237 *
238 *
239 *
240 *
241 *
242 *
243 *
244 *
245 *
246 *
247 *
248 *
249 *
250 *
251 *
252 *
253 *
254 *
255 *
256 *
257 *
258 *
259 *
260 *
261 *
262 *
263 *
264 *

```

```

353 LIST NEXT listrecs curr_num, "curr_name, ", dept_code, " phone_no
354 SET HEADING ON
355 RETURN
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```



```

87 (recall_3 < GORONTH(DATE(), -1)*5) ;
88 .AND. (EMPTY(recall_4)))
89 CNT = 4*
90 DO goreal1s
91 SET FILTER TO recall3 at beginning of 13th month since last_t2
92 SET FILTER TO (last_t2 < GORONTH(DATE(), -12)) ;
93 .AND. (members.recall_2 < GORONTH(DATE(), -1)*5) ;
94 .AND. (NOT EMPTY(recall_2)) .AND. (EMPTY(recall_3)))
95 CNT = 3*
96 DO goreal1s
97 SET FILTER TO recall2 at beginning of 12th month since last_t2
98 SET FILTER TO (last_t2 < GORONTH(DATE(), -11)) ;
99 .AND. (members.recall_1 < GORONTH(DATE(), -1)*5) ;
100 .AND. (NOT EMPTY(recall_1)) .AND. (EMPTY(recall_2)))
101 CNT = 2*
102 DO goreal1s
103 SET FILTER TO recall1 at beginning of 11th month since last_t2
104 SET FILTER TO (last_t2 < GORONTH(DATE(), -10)) ;
105 .AND. (EMPTY(recall_1)))
106 CNT = 1*
107 DO goreal1s
108 aerolntent = .F.
109 ENDDO
110 ON ESCAPE
111 SET CURSOR ON
112 ENDIF
113
114 ON KEY * 315
115 SET FILTER TO
116 SET TEXTFILER TO
117 CASE choice = "2" .OR. choice = "9"
118 SET CURSOR OFF
119 SET COLOR TO W/M
120 DO rec_lists
121 CASE choice = "3" .OR. choice = "V"
122 SET CURSOR OFF
123 SET COLOR TO W/M
124 @ 24,0 SAY "WORKING ..." COLOR W/M
125 DO nos_crec
126 ENDCASE
127 ENDDO
128
129
130
131
132
133
134
135 PROCEDURE recalls_menu
136 PRIVATE COL
137
138 @ 0,40 SAY DATE() COLOR W/M
139 @ 0,48 SAY " " COLOR W/M
140 @ ---Center the menu heading.
141 menuhdg = "P T A R S R E C A L L S M E N U"
142 COL = (80 - LEN(menuhdg)) / 2
143 @ 1,0 TO 10,79 DOUBLE COLOR G/M
144 @ 3,0 SAY " "
145 @ 3,40 SAY " "
146 @ 2,COL SAY menuhdg COLOR GR/M
147 SET COLOR TO W/M,N/W
148 @ 5,3 SAY " (F1) for help"
149 @ 5,8 SAY "F1" COLOR GR/M
150 @ 6,3 SAY "Alt+F1) for functions"
151 @ 6,4 SAY "Alt+F1" COLOR GR/M
152 COL = 34
153 @ 5,COL SAY "0. Exit to main menu"
154 @ 5,COL + 3 SAY "E" COLOR GR/M
155 @ 6,COL SAY "1. Print recalls"
156 @ 6,COL + 3 SAY "P" COLOR GR/M
157 @ 7,COL SAY "2. Print most recent recall list"
158 @ 7,COL + 4 SAY "R" COLOR GR/M
159 @ 8,COL SAY "3. View/edit recall dates"
160 @ 8,COL + 3 SAY "V" COLOR GR/M
161 @ 10,33 SAY "select : "
162 SET CURSOR ON
163 @ 10,42 SAY ""
164 RETURN
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
```

```

175 *1
176
177 PROCEDURE rec_lists
178 IF NOT LASTKEY() = 27      &&f not escape key pressed
179   DEFINE POPUP rec_lists FROM 4,48 TO 10,79 TITLE "Select"
180   MESSAGE "Scroll or press highlighted letter to select list"
181   "to print" COLOR W/BG,W/BG,B/BG,B/BG,W/W,B/BG/BG
182   DEFINE BAR 1 OF rec_lists PROMPT = Recall V1
183   DEFINE BAR 2 OF rec_lists PROMPT = Recall V2
184   DEFINE BAR 3 OF rec_lists PROMPT = Recall V3
185   DEFINE BAR 4 OF rec_lists PROMPT = Recall V4
186   DEFINE BAR 5 OF rec_lists PROMPT = VALL
187   ON SELECTION POPUP rec_lists DO dactopop
188   &&pop
189   @ 10,33 SAY " " COLOR G/W
190   ACTIVATE POPUP rec_lists
191   STORE .T. TO morint
192   DO CASE
193     CASE morprompt = " Recall 1 "
194       rectype = "recall1.lst"
195       DO prnchk
196       DO recrn WITH rectype
197       EFFECT
198     CASE morprompt = " Recall 2 "
199       rectype = "recall2.lst"
200       DO prnchk
201       DO recrn WITH rectype
202       EFFECT
203     CASE morprompt = " Recall 3 "
204       rectype = "recall3.lst"
205       DO prnchk
206       DO recrn WITH rectype
207       EFFECT
208     CASE morprompt = " Recall 4 "
209       rectype = "recall4.lst"
210       DO prnchk
211       DO recrn WITH rectype
212       EFFECT
213     CASE morprompt = " All "
214       counter = 0
215       DO prnchk
216       ON ERROR DO prnprob
217       * Problem
218       DO WHILE (counter < 4) AND, morint
219         counter = counter + 1
220         rectype = "recall" + LTRIM(STR(counter)) + ".lst"
221         DO recrn WITH rectype
222         ENDDO
223       CLEAR
224       EFFECT
225     ENDCASE
226   RELEASE POPUP rec_lists
227   RETURN
228 *1
229
230 PROCEDURE recrn
231
232 Called by: REC_LISTS
233
234 CALLS: PRNPROB
235       PRNSTOP
236
237 Other Files: SRECTYPE
238
239
240 PROCEDURE recrn
241 PARAMETER rectype
242 IF NOT LASTKEY() = 27      &&f not escape key pressed
243   n ----Escape routine
244   CLEAR
245   SET ESCAPE ON
246   STORE .T. TO morint,morintcnt
247   STORE .T. TO morint
248   ON ERROR DO prnprob
249   * Problem
250   DO WHILE morprompt AND, morint
251     n ----Print recall list
252     CLEAR
253     IF morint
254       @ 0,24 SAY "Printing " + rectype + " listing ..." COLOR W/W
255       @ 1,24 SAY "Press (Esc) to abort" COLOR W/W
256     ENDIF
257     TYPE rectype TO PRINTER
258     morintcnt = .F.
259   ENDDO

```



```

352 IF prntat = "Y"
353 W ---Escape routine
354 SET CURSOR OFF
355 CLEAR
356 SET ESCAPE ON
357 STORE .1. TO aprnt.escpt
358 ON ERROR DO aprnt.escpt
359 * r problem
360 ON ESCAPE DO aprnt.escpt
361 DO WHILE aprnt.escpt AND aprnt
362 @ 10.24 SAY "Sending report to printer ..." COLOR W/BL
363 @ 11.24 SAY "Press (Esc) to abort" COLOR W/BL
364 DO RETURN
365 REPORT FORM rptname TO PRINTER NOCONSOLE
366 aprnt.escpt = .F.
367 ENDOO
368 ON ERROR
369 ON ESCAPE
370 SET CURSOR ON
371 ENDOF
372 ON KEY = 315
373 RETURN
374 W, ED, NPS_REPO.PRG

```

```

244 PROCEDURE classpop
245 IF NOT LASTKEY() = 27
246 DEFINE POPUP classpop FROM 4.55 TO 12.67 TITLE "Select";
247 MESSAGE "Scroll" or press highlighted letter to select Class ";
248 "to print" COLOR W/BL,W/BL,BG,BG,W/BL,W/BL,W/BL,W/BL,W/BL,W/BL
249 DEFINE BAR 1 OF classpop PROMPT "Class \1"
250 DEFINE BAR 2 OF classpop PROMPT "Class \2"
251 DEFINE BAR 3 OF classpop PROMPT "Class \3"
252 DEFINE BAR 4 OF classpop PROMPT "Class \4"
253 DEFINE BAR 5 OF classpop PROMPT "Class \5"
254 DEFINE BAR 6 OF classpop PROMPT "Class \6"
255 DEFINE BAR 7 OF classpop PROMPT "Class \7"
256 ON SELECTION POPUP classpop DO deactpop
257 aprnt.escpt = .F.
258 @ 14.33 SAY " " COLOR G/Y
259 ACTIVATE POPUP classpop
260 DO CASE
261 CASE aprnt.escpt = "Class 1"
262 SET FILTER TO class = 1
263 CASE aprnt.escpt = "Class 2"
264 SET FILTER TO class = 2
265 CASE aprnt.escpt = "Class 3"
266 SET FILTER TO class = 3
267 CASE aprnt.escpt = "Class 4"
268 SET FILTER TO class = 4
269 CASE aprnt.escpt = "Class 1 & 2"
270 SET FILTER TO (class = 1) .OR. (class = 2)
271 CASE aprnt.escpt = "Class 3 & 4"
272 SET FILTER TO (class = 3) .OR. (class = 4)
273 CASE aprnt.escpt = "All Classes"
274 SET FILTER TO
275 ENDOF
276 RELEASE POPUP classpop
277 ENDOF
278 RETURN
279 *****
280 *****
281 *****
282 *****
283 *****
284 *****
285 *****
286 *****
287 *****
288 *****
289 *****
290 *****
291 *****
292 *****
293 *****
294 *****
295 *****
296 *****
297 *****
298 *****
299 *****
300 *****
301 *****
302 *****
303 *****
304 *****
305 *****
306 *****
307 *****
308 *****
309 *****
310 *****
311 *****
312 *****
313 *****
314 *****
315 *****
316 *****
317 *****
318 *****
319 *****
320 *****
321 *****
322 *****
323 *****
324 *****
325 *****
326 *****
327 *****
328 *****
329 *****
330 *****
331 *****
332 *****
333 *****
334 *****
335 *****
336 *****
337 *****
338 *****
339 *****
340 *****
341 *****
342 *****
343 *****
344 *****
345 *****
346 *****
347 *****
348 *****
349 *****
350 *****
351 *****

```

```

90 0 1,0 TO 12,79 DOUBLE COLOR G/W
91 0 3,0 SAY " "
92 0 3,0 SAY " "
93 0 2,0 COL SAY N499 COLOR GR/W
94 SET COLOR TO W/W/N,N/W
95 COL = 34
96 0 5,3 SAY " "
97 0 5,3 SAY "F1" COLOR GR/W
98 0 6,3 SAY "Alt+F1" for function"
99 0 6,4 SAY "Alt+F1" COLOR GR/W
100 0 5,0 COL SAY "0. Exit to main menu"
101 0 5,0 COL SAY "1. Backup utilities"
102 0 6,0 COL SAY "1. Backup utilities"
103 0 7,0 COL SAY "2. Change password"
104 0 7,0 COL SAY "3. Pack database(s)"
105 0 8,0 COL SAY "3. Pack database(s)"
106 0 8,0 COL SAY "4. Change date/time"
107 0 9,0 COL SAY "4. Change date/time"
108 0 9,0 COL + 4 SAY "N" COLOR GR/W
109 0 10,0 COL SAY "5. default printer"
110 0 10,0 COL + 5 SAY "F" COLOR GR/W
111 0 12,33 SAY "select : "
112 SET CURSOR ON
113 0 12,42 SAY ""
114 RETURN
115
116
117 Procedure, PRDEFALT
118
119 Called by: NPS_UTIL.PRG
120
121
122 Calls: DEACTOP
123
124 (procedure in NPS_PROC.PRG)
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
```

[illegible]


```

245 n=1 CALL GETKEY (procedure in NPS_PROC.PRG)
246 n=1 NPS_PACK.PRG
247 n=1
248 =====
249 PROCEDURE warning
250 PROMPTERS orig1,orignd - COLOR G/N
251 0 12.31 SAY
252 0 13.0 CLEAR TO 24,79
253 0 16, 3 TO 15,76 COLOR W/RB
254 0 17, 4 SAY "Packing the database permanently removes records ";
255 "marked for deletion"
256 0 17,27 SAY "permanently" COLOR W/RB
257 0 18, 4 SAY "and improve the performance of the database. It ";
258 "is not undoable." - COLOR W/RB
259 SET CURSOR ON
260 21,18 SAY "Do you want to pack the database now? (y/n)" - COLOR W/N
261 DO key1 WITH choice;"n"=returnkey
262 IF choice = "y"
263 RETURN
264 ENDIF
265 DO nps_pack WITH orig1,orignd
266 RETURN
267 n=1 =====
268 n=1 Procedure: NPS_CPAS
269 n=1 Called by: NPS_UTIL.PRG
270 n=1
271 n=1 CALL: HLP
272 n=1 (procedure in NPS_PROC.PRG)
273 n=1 PASSTRAIN
274 n=1 (procedure in NPS_PROC.PRG)
275 n=1 =====
276 n=1 =====
277 PROCEDURE nps_cpas
278 n=1 --- Change password
279 PRIVATE newpass,verify,colget,colkey,amax,c
280 n=1 ---Using NPS_MISC.DBF
281 SELECT E
282 CLEAR
283 0 0,40 SAY DATE() COLOR N/M
284 0 0,48 SAY " - COLOR N/M
285 n=1 --- Center the fore heading
286 menuhdg = "P A S S W O R D C H A N G E U T I L I T Y"
287 COL = (80 - LEN(menuhdg)) / 2
288 0 1,0 TO 11,79 DOUBLE COLOR G/N
289 0 3, 0 SAY " - COLOR G/N
290 0 3,40 SAY " - COLOR G/N
291 0 2,COL SAY menuhdg COLOR GR/N
292 SET COLOR TO W/N,N/M
293 newpass = SPACE(6)
294 verify = SPACE(6)
295 ON KEY LABEL f1 DO hlp WITH "Paaa"
296 0 5, 3 SAY " <F1> for help"
297 0 5, 8 SAY "F1" COLOR GR/N
298 0 6, 3 SAY "Alt+F1" for functions"
299 0 6, 4 SAY "Alt+F1" COLOR GR/N
300 0 7, 3 SAY " <Esc> to abort"
301 0 7, 7 SAY "Esc" COLOR GR/N
302 0 5,34 SAY "Enter new password."
303 0 5,55 SAY " - COLOR N/M
304 colget = 64
305 colkey = 54
306 newpass = ""
307 SET CURSOR ON
308 FOR i = 1 TO 6
309 amax = SPACE(1)
310 0 5,colget + 1 GET amax
311 VALID (UPPER(verify)) = "ABCDEFHIJKLMNOPQRSTUVWXYZ"
312 ERROR ("Only alpha characters allowed. Continue entry ...")
313 COLOR ,1
314 READ
315 IF LASTKEY() = 27
316 SET CURSOR OFF
317 WAIT "Password change aborted." WINDOW TIMEOUT 1
318 SET CURSOR ON
319 EXIT
320 ENDIF
321 0 5,colkey + 1 SAY " -
322 newpass = newpass+amax
323 CLEAR GETS
324 EMPUR
325 IF newpass = " - OR, (LEN(ALLTRIM(newpass)) < 6)
326 RETURN
327 ELSE
328 77 CHR(7)
329 0 7,34 SAY "Verify new password."
330 0 7,55 SAY " - COLOR N/M
331 verify = ""
332 FOR i = 1 TO 6

```

```

353 amax = SPACE(1)
354 0 7,colget + 1 GET amax COLOR ,1
355 READ
356 IF LASTKEY() = 27
357 SET CURSOR OFF
358 WAIT "Password change aborted." WINDOW TIMEOUT 1
359 SET CURSOR ON
360 CLEAR
361 EXIT
362 ENDIF
363 0 7,colkey + 1 SAY " -
364 verify = newpass+verify+amax
365 CLEAR GETS
366 EMPFOR
367 IF LASTKEY() = 27
368 SET CURSOR ON
369 EXIT
370 ENDIF
371 SET CURSOR OFF
372 IF newpass = verify
373 REPLACE password WITH password(newpass,6,1,1)
374 0 9,34 SAY "Verification successful."
375 WAIT -- TIMEOUT 1
376 ELSE
377 77 CHR(7)
378 WAIT "Verification failure. Original password remains in effect."
379 WINDOW TIMEOUT 1
380 ENDIF
381 SET CURSOR ON
382 RETURN
383 n=1 EOF, NPS_UTIL.PRG

```

```

1  *****
2  1. Procedure file, C:\VARS\NPS_BU.PRG
3  2.
4  3.
5  4.
6  5.
7  6.
8  7.
9  8.
10 9.
11 10.
12 11.
13 12.
14 13.
15 14.
16 15.
17 16.
18 17.
19 18.
20 19.
21 20.
22 21.
23 22.
24 23.
25 24.
26 25.
27 26.
28 27.
29 28.
30 29.
31 30.
32 31.
33 32.
34 33.
35 34.
36 35.
37 36.
38 37.
39 38.
40 39.
41 40.
42 41.
43 42.
44 43.
45 44.
46 45.
47 46.
48 47.
49 48.
50 49.
51 50.
52 51.
53 52.
54 53.
55 54.
56 55.
57 56.
58 57.
59 58.
60 59.
61 60.
62 61.
63 62.
64 63.
65 64.
66 65.
67 66.
68 67.
69 68.
70 69.
71 70.
72 71.
73 72.
74 73.
75 74.
76 75.
77 76.
78 77.
79 78.
80 79.
81 80.
82 81.
83 82.
84 83.
85 84.
86 85.
87 86.

```

```

88 0 1.0 TO 11.79 DOUBLE COLOR G/N
89 0 3.0 SAY "
90 0 3.40 SAY "
91 0 2.0L SAY hds COLOR GR/N
92 0L " 34
93 0 5. 3 SAY "
94 0 5. 0 SAY "F1" COLOR GR/N
95 0 4. 3 SAY "Alt+F1" for function
96 0 4. 4 SAY "Alt+F1" COLOR GR/N
97 0 4. 4 SAY "Alt+F1" COLOR GR/N
98 IF BU = "F.
99 0 5.0L SAY "0. Exit to utilities menu"
100 0 5.0L + 3 SAY "E" COLOR GR/N
101 0 5.0L SAY "0. Quit"
102 0 5.0L + 3 SAY "0" COLOR GR/N
103 0 5.0L + 3 SAY "0" COLOR GR/N
104 ENDIF
105 0 4.0L SAY "1. Backup database file(s)".
106 0 4.0L + 3 SAY "2" COLOR GR/N
107 0 7.0L SAY "2. List files".
108 0 7.0L + 3 SAY "3" COLOR GR/N
109 0 0.0L SAY "3. Format floppy disk (A:)"
110 0 0.0L + 3 SAY "F" COLOR GR/N
111 0 9.0L SAY "4. Restore backup(s)".
112 0 9.0L + 3 SAY "R" COLOR GR/N
113 0 11.33 SAY "select i"
114 0 11.42 SAY "
115 SET CURSOR ON
116 RETURN
117 *****
118 *****
119 *****
120 *****
121 *****
122 *****
123 *****
124 *****
125 *****
126 *****
127 *****
128 *****
129 *****
130 *****
131 *****
132 *****
133 *****
134 *****
135 *****
136 *****
137 *****
138 *****
139 *****
140 *****
141 *****
142 *****
143 *****
144 *****
145 *****
146 *****
147 *****
148 *****
149 *****
150 *****
151 *****
152 *****
153 *****
154 *****
155 *****
156 *****
157 *****
158 *****
159 *****
160 *****
161 *****
162 *****
163 *****
164 *****
165 *****
166 *****
167 *****
168 *****
169 *****
170 *****
171 *****
172 *****
173 *****
174 *****
175 *****

```

```

176 *****
177 *****
178 *****
179 *****
180 *****
181 *****
182 *****
183 *****
184 *****
185 *****
186 *****
187 *****
188 *****
189 *****
190 *****
191 *****
192 *****
193 *****
194 *****
195 *****
196 *****
197 *****
198 *****
199 *****
200 *****
201 *****
202 *****
203 *****
204 *****
205 *****
206 *****
207 *****
208 *****
209 *****
210 *****
211 *****
212 *****
213 *****
214 *****
215 *****
216 *****
217 *****
218 *****
219 *****
220 *****
221 *****
222 *****
223 *****
224 *****
225 *****
226 *****
227 *****
228 *****
229 *****
230 *****
231 *****
232 *****
233 *****
234 *****
235 *****
236 *****
237 *****
238 *****
239 *****
240 *****
241 *****
242 *****
243 *****
244 *****
245 *****
246 *****
247 *****
248 *****
249 *****
250 *****
251 *****
252 *****
253 *****
254 *****
255 *****
256 *****
257 *****
258 *****
259 *****
260 *****
261 *****
262 *****
263 *****

```

```

244 SET ORDER TO 1
245 orig = "MEMBERS.DBF"
246 DO CASE
247     buf = "REN_BU.DBF"
248     DO def bu WITH ROW,orig,buf
249     CASE bukind = "F"
250         buf = "A.MEMBERS.DBF"
251     DO bu_single WITH orig,buf
252     ENDCASE
253 CASE morment = "activity"
254     SELECT 2
255     SET ORDER TO 1
256     orig = "ACTIVITY.DBF"
257     DO CASE
258         buf = "H"
259         DO def bu WITH ROW,orig,buf
260         CASE bukind = "F"
261             buf = "A.ACTIVITY.DBF"
262         DO bu_single WITH orig,buf
263         ENDCASE
264 CASE morment = "Curriculum"
265     SELECT 3
266     SET ORDER TO 1
267     orig = "CURRICUL.DBF"
268     DO CASE
269         buf = "H"
270         DO def bu WITH ROW,orig,buf
271         CASE bukind = "F"
272             buf = "A.CURRICUL.DBF"
273         DO bu_single WITH orig,buf
274         ENDCASE
275 CASE morment = "Director"
276     SELECT 4
277     SET ORDER TO 1
278     orig = "DIRECTOR.DBF"
279     DO CASE
280         buf = "DIR_BU.DBF"
281         DO def bu WITH ROW,orig,buf
282         CASE bukind = "F"
283             buf = "A.CURRICUL.DBF"
284         DO bu_single WITH orig,buf
285         ENDCASE
286 CASE morment = "Director"
287     SELECT 4
288     SET ORDER TO 1
289     orig = "DIRECTOR.DBF"
290     DO CASE
291         buf = "H"
292         DO def bu WITH ROW,orig,buf
293         CASE bukind = "F"
294             buf = "A.CURRICUL.DBF"
295         DO bu_single WITH orig,buf
296         ENDCASE
297 CASE morment = "Director"
298     SELECT 4
299     SET ORDER TO 1
300     orig = "DIRECTOR.DBF"
301     DO CASE
302         buf = "H"
303         DO def bu WITH ROW,orig,buf
304         CASE bukind = "F"
305             buf = "A.DIRECTOR.DBF"
306         DO bu_single WITH orig,buf
307         ENDCASE
308 CASE morment = "Director"
309     SELECT 4
310     SET ORDER TO 1
311     orig = "DIRECTOR.DBF"
312     DO CASE
313         buf = "H"
314         DO def bu WITH ROW,orig,buf
315         CASE bukind = "F"
316             buf = "A.DIRECTOR.DBF"
317         DO bu_single WITH orig,buf
318         ENDCASE
319 CASE morment = "Director"
320     SELECT 4
321     SET ORDER TO 1
322     orig = "DIRECTOR.DBF"
323     DO CASE
324         buf = "H"
325         DO def bu WITH ROW,orig,buf
326         CASE bukind = "F"
327             buf = "A.DIRECTOR.DBF"
328         DO bu_single WITH orig,buf
329         ENDCASE
330 CASE morment = "Director"
331     SELECT 4
332     SET ORDER TO 1
333     orig = "DIRECTOR.DBF"
334     DO CASE
335         buf = "H"
336         DO def bu WITH ROW,orig,buf
337         CASE bukind = "F"
338             buf = "A.DIRECTOR.DBF"
339         DO bu_single WITH orig,buf
340         ENDCASE
341 CASE morment = "Director"
342     SELECT 4
343     SET ORDER TO 1
344     orig = "DIRECTOR.DBF"
345     DO CASE
346         buf = "H"
347         DO def bu WITH ROW,orig,buf
348         CASE bukind = "F"
349             buf = "A.DIRECTOR.DBF"
350         DO bu_single WITH orig,buf
351         ENDCASE

```



```

527 DIRECTORY ON A: LIKE *.
528 WAIT
529 ENDCASE
530 PROCEDURE RESTOR
531 RELEASE POPUP list_files
532 SET CURSOR ON
533 RETURN
534 *****
535 *****
536 *****
537 *****
538 *****
539 *****
540 *****
541 *****
542 *****
543 *****
544 *****
545 *****
546 *****
547 *****
548 *****
549 *****
550 *****
551 *****
552 *****
553 *****
554 *****
555 *****
556 *****
557 *****
558 *****
559 *****
560 *****
561 *****
562 *****
563 *****
564 *****
565 *****
566 *****
567 *****
568 *****
569 *****
570 *****
571 *****
572 *****
573 *****
574 *****
575 *****
576 *****
577 *****
578 *****
579 *****
580 *****
581 *****
582 *****
583 *****
584 *****
585 *****
586 *****
587 *****
588 *****
589 *****
590 *****
591 *****
592 *****
593 *****
594 *****
595 *****
596 *****
597 *****
598 *****
599 *****
600 *****
601 *****
602 *****
603 *****
604 *****
605 *****
606 *****
607 *****
608 *****
609 *****
610 *****
611 *****

```

```

612 RETURN
613 *****
614 *****
615 *****
616 *****
617 *****
618 *****
619 *****
620 *****
621 *****
622 *****
623 *****
624 *****
625 *****
626 *****
627 *****
628 *****
629 *****
630 *****
631 *****
632 *****
633 *****
634 *****
635 *****
636 *****
637 *****
638 *****
639 *****
640 *****
641 *****
642 *****
643 *****
644 *****
645 *****
646 *****
647 *****
648 *****
649 *****
650 *****
651 *****
652 *****
653 *****
654 *****
655 *****
656 *****
657 *****
658 *****
659 *****
660 *****
661 *****
662 *****
663 *****
664 *****
665 *****
666 *****
667 *****
668 *****
669 *****
670 *****
671 *****
672 *****
673 *****
674 *****
675 *****
676 *****
677 *****
678 *****
679 *****
680 *****
681 *****
682 *****
683 *****
684 *****
685 *****
686 *****
687 *****
688 *****
689 *****
690 *****
691 *****
692 *****
693 *****
694 *****
695 *****
696 *****
697 *****
698 *****

```

```

699 *****
700 *****
701 *****
702 *****
703 *****
704 *****
705 *****
706 *****
707 *****
708 *****
709 *****
710 *****
711 *****
712 *****
713 *****
714 *****
715 *****
716 *****
717 *****
718 *****
719 *****
720 *****
721 *****
722 *****
723 *****
724 *****
725 *****
726 *****
727 *****
728 *****
729 *****
730 *****
731 *****
732 *****
733 *****
734 *****
735 *****
736 *****
737 *****
738 *****
739 *****
740 *****
741 *****
742 *****
743 *****
744 *****
745 *****
746 *****
747 *****
748 *****
749 *****
750 *****
751 *****
752 *****
753 *****
754 *****
755 *****
756 *****
757 *****
758 *****
759 *****
760 *****
761 *****
762 *****
763 *****
764 *****
765 *****
766 *****
767 *****
768 *****
769 *****
770 *****
771 *****
772 *****
773 *****
774 *****
775 *****
776 *****
777 *****
778 *****
779 *****

```

```

848 GOTO BOTTOM
849 rcd_cnt = RECDNT()
850 0 15.0 CLEAR TO 23.79
851 0 15.0 SAY (Raskoring "i + origfile + [" from floppy disk backup "] + s
852 0 16.0 SAY "Continuing ..."
853 0 14.0 SAY "Continuing ..."
854 SET TALK ON
855 COPY TO &tempfile
856 SET TALK OFF
857 WAIT -- TIMEOUT i
858 USE
859 n ---Check if restoration is OK
860 lefileok = ".F."
861 IF FILE(&tempfile)
862 USE &tempfile
863 GOTO BOTTOM
864 0 15.0 SAY "Record count from original and backup must match
865 lefileok = (rcd_cnt = RECDNT())
866 USE
867 ENDDIF
868 IF ( ! lefileok)
869 n ---Temp file was not created or has incorrect record count
870 WAIT(7)
871 77 CHRT(7)
872 0 15.0 SAY "Restoration failed - WINDOW TIMEOUT 2
873 IF FILE(&tempfile)
874 ERASE &tempfile
875 ENDDIF
876 RETURN
877 ENDDIF
878 ERASE &origfile
879 REMAKE &tempfile TO &origfile
880 ERASE &tempfile
881 0 15.0 CLEAR TO 23.79
882 0 15.0 SAY (floppy disk backup "] + &tempfile + [" does not exist,i
883 WAIT -- WINDOW TIMEOUT 2
884 ENDDIF
885 9005 RETURN
886 9006 n, EOF, MP5_BU, PRG

```



```

1  * *****
2  * Procedure file: C:\VTAMSWPS_PACK.PRG
3  *
4  * Proc & Fnct: PACK
5  *
6  *
7  * Set by: PACK_CHECK
8  *   (procedure in NPSDC.PRG)
9  *   (procedure in NPS_UTIL.PRG)
10 *   (procedure in NPS_UTIL.PRG)
11 *   (procedure in NPS_UTIL.PRG)
12 *   (procedure in NPS_UTIL.PRG)
13 *   (procedure in NPS_UTIL.PRG)
14 *   (procedure in NPS_UTIL.PRG)
15 *   (procedure in NPS_UTIL.PRG)
16 *   (procedure in NPS_UTIL.PRG)
17 *   (procedure in NPS_UTIL.PRG)
18 *   (procedure in NPS_UTIL.PRG)
19 *   (procedure in NPS_UTIL.PRG)
20 *   (procedure in NPS_UTIL.PRG)
21 *   (procedure in NPS_UTIL.PRG)
22 *   (procedure in NPS_UTIL.PRG)
23 *   (procedure in NPS_UTIL.PRG)
24 *   (procedure in NPS_UTIL.PRG)
25 *   (procedure in NPS_UTIL.PRG)
26 *   (procedure in NPS_UTIL.PRG)
27 *   (procedure in NPS_UTIL.PRG)
28 *   (procedure in NPS_UTIL.PRG)
29 *   (procedure in NPS_UTIL.PRG)
30 *   (procedure in NPS_UTIL.PRG)
31 *   (procedure in NPS_UTIL.PRG)
32 *   (procedure in NPS_UTIL.PRG)
33 *   (procedure in NPS_UTIL.PRG)
34 *   (procedure in NPS_UTIL.PRG)
35 *   (procedure in NPS_UTIL.PRG)
36 *   (procedure in NPS_UTIL.PRG)
37 *   (procedure in NPS_UTIL.PRG)
38 *   (procedure in NPS_UTIL.PRG)
39 *   (procedure in NPS_UTIL.PRG)
40 *   (procedure in NPS_UTIL.PRG)
41 *   (procedure in NPS_UTIL.PRG)
42 *   (procedure in NPS_UTIL.PRG)
43 *   (procedure in NPS_UTIL.PRG)
44 *   (procedure in NPS_UTIL.PRG)
45 *   (procedure in NPS_UTIL.PRG)
46 *   (procedure in NPS_UTIL.PRG)
47 *   (procedure in NPS_UTIL.PRG)
48 *   (procedure in NPS_UTIL.PRG)
49 *   (procedure in NPS_UTIL.PRG)
50 *   (procedure in NPS_UTIL.PRG)
51 *   (procedure in NPS_UTIL.PRG)
52 *   (procedure in NPS_UTIL.PRG)
53 *   (procedure in NPS_UTIL.PRG)
54 *   (procedure in NPS_UTIL.PRG)
55 *   (procedure in NPS_UTIL.PRG)
56 *   (procedure in NPS_UTIL.PRG)
57 *   (procedure in NPS_UTIL.PRG)
58 *   (procedure in NPS_UTIL.PRG)
59 *   (procedure in NPS_UTIL.PRG)
60 *   (procedure in NPS_UTIL.PRG)
61 *   (procedure in NPS_UTIL.PRG)
62 *   (procedure in NPS_UTIL.PRG)
63 *   (procedure in NPS_UTIL.PRG)
64 *   (procedure in NPS_UTIL.PRG)
65 *   (procedure in NPS_UTIL.PRG)
66 *   (procedure in NPS_UTIL.PRG)
67 *   (procedure in NPS_UTIL.PRG)
68 *   (procedure in NPS_UTIL.PRG)
69 *   (procedure in NPS_UTIL.PRG)
70 *   (procedure in NPS_UTIL.PRG)
71 *   (procedure in NPS_UTIL.PRG)
72 *   (procedure in NPS_UTIL.PRG)
73 *   (procedure in NPS_UTIL.PRG)
74 *   (procedure in NPS_UTIL.PRG)
75 *   (procedure in NPS_UTIL.PRG)
76 *   (procedure in NPS_UTIL.PRG)
77 *   (procedure in NPS_UTIL.PRG)
78 *   (procedure in NPS_UTIL.PRG)
79 *   (procedure in NPS_UTIL.PRG)
80 *   (procedure in NPS_UTIL.PRG)
81 *   (procedure in NPS_UTIL.PRG)
82 *   (procedure in NPS_UTIL.PRG)
83 *   (procedure in NPS_UTIL.PRG)
84 *   (procedure in NPS_UTIL.PRG)
85 *   (procedure in NPS_UTIL.PRG)

```

```

86 ENDIF
87 *****
88 USE ORIGINAL FILE
89 *****
90 DO PACK
91 *****
92 USE NOT deleted = 0
93 *****
94 USE NOT deleted = 0
95 *****
96 USE NOT deleted = 0
97 *****
98 USE NOT deleted = 0
99 *****
100 *****
101 *****
102 *****
103 *****
104 *****
105 *****
106 *****
107 *****
108 *****
109 *****
110 *****
111 *****
112 *****
113 *****
114 *****
115 *****
116 *****
117 *****
118 *****
119 *****
120 *****
121 *****
122 *****
123 *****
124 *****
125 *****
126 *****
127 *****
128 *****
129 *****
130 *****
131 *****
132 *****
133 *****
134 *****
135 *****
136 *****
137 *****
138 *****
139 *****
140 *****
141 *****
142 *****
143 *****
144 *****
145 *****
146 *****
147 *****
148 *****
149 *****
150 *****
151 *****
152 *****
153 *****
154 *****
155 *****
156 *****

```

NPS PROC.PRG 1 of 7

NPS PROC.PRG 1 of 7

NPS PROC.PRG 1 of 7

```

264 RETURN
265 =====
266
267 Procedure: BAR1
268
269 Called by: FUNC_LIST
270
271 Calls: HLP
272
273
274 PROCEDURE bar1
275 DO HIP WITH "Func"
276 DEACTIVATE POPUP func
277 RETURN
278
279 =====
280
281 Procedure: BAR2
282
283 Called by: FUNC_LIST
284
285 Calls: POPCAL
286
287 PROCEDURE bar2
288 DO POPCAL
289 DEACTIVATE POPUP func
290 RETURN
291
292 =====
293
294 Procedure: BAR3
295
296 Called by: FUNC_LIST
297
298 Calls: POPTRIS.PRG
299
300 PROCEDURE bar3
301 DO POPTRIS
302 DEACTIVATE POPUP func
303 RETURN
304
305 =====
306
307 Procedure: BAR4
308
309 Called by: FUNC_LIST
310
311 Calls: ABOUT
312
313 PROCEDURE bar4
314 DO ABOUT
315 DEACTIVATE POPUP func
316 RETURN
317
318 =====
319
320 Procedure: VIEWBROW
321
322 Called by: NPS_BROW.PRG
323
324 Calls: DEACTPOP
325
326 PROCEDURE viewbrow
327
328 ---Change screen display mode
329 SET COLOR TO W/N,N/M
330 ---Test for EGA or VGA to do View mode
331 IF ( ("EGA" & SYS( 206 ) ) .OR. "VGA" & SYS( 206 ) )
332 77 CHR(7)
333 WAIT "Sorry. (Mode requires EGA/VGA video adapter." WINDOW TIMEOUT 2
334 RETURN ( ... )
335
336 ENDIF
337
338 MESSAGE "Scroll or press highlighted letter to select a View mode";
339 COLOR W/B,M/86,B/86,G/86,B/W,N/M/R/86
340 DEFINE BAR 1 OF mode PROMPT "EGA\43"
341 DEFINE BAR 2 OF mode PROMPT "EGA\43"
342 DEFINE BAR 3 OF mode PROMPT "VGA\30"
343 ON SELECTION POPUP mode DO deactpop
344
345 DEACTIVATE POPUP mode
346 DO CASE
347 CASE omet = "EGA25"
348 CLEAR
349 reanetop = RECHO(1)
350 SET DISPLAY TO EGA25
351

```

```

352 CASE reanetop = "EGA43"
353 CLEAR
354 reanetop = RECHO(1)
355 SET DISPLAY TO EGA43
356
357 CASE reanetop = "VGA25"
358 IF ( ("VGA" & SYS( 206 ) )
359 77 CHR(7)
360 RETURN
361
362 ELSE
363 CLEAR
364 reanetop = RECHO(1)
365 SET DISPLAY TO VGA25
366
367
368 CASE reanetop = "VGA50"
369 IF ( ("VGA" & SYS( 206 ) )
370 77 CHR(7)
371 WAIT "Sorry, VGA video adapter required." WINDOW TIMEOUT 2
372 RETURN
373
374 ELSE
375 CLEAR
376 SET DISPLAY TO VGA50
377
378
379 ENDIF
380
381 CASE reanetop = "VGA50"
382 IF ( ("VGA" & SYS( 206 ) )
383 77 CHR(7)
384 RETURN
385
386 ELSE
387 CLEAR
388 SET DISPLAY TO VGA50
389
390
391 ENDIF
392
393 CASE reanetop = "VGA50"
394 IF ( ("VGA" & SYS( 206 ) )
395 77 CHR(7)
396 RETURN
397
398 ELSE
399 CLEAR
400 SET DISPLAY TO VGA50
401
402
403 ENDIF
404
405 CASE reanetop = "VGA50"
406 IF ( ("VGA" & SYS( 206 ) )
407 77 CHR(7)
408 RETURN
409
410 ELSE
411 CLEAR
412 SET DISPLAY TO VGA50
413
414
415 ENDIF
416
417 CASE reanetop = "VGA50"
418 IF ( ("VGA" & SYS( 206 ) )
419 77 CHR(7)
420 RETURN
421
422 ELSE
423 CLEAR
424 SET DISPLAY TO VGA50
425
426
427 ENDIF
428
429 CASE reanetop = "VGA50"
430 IF ( ("VGA" & SYS( 206 ) )
431 77 CHR(7)
432 RETURN
433
434 ELSE
435 CLEAR
436 SET DISPLAY TO VGA50
437
438
439 ENDIF
440
441 CASE reanetop = "VGA50"
442 IF ( ("VGA" & SYS( 206 ) )
443 77 CHR(7)
444 RETURN
445
446 ELSE
447 CLEAR
448 SET DISPLAY TO VGA50
449
450
451 ENDIF
452
453 CASE reanetop = "VGA50"
454 IF ( ("VGA" & SYS( 206 ) )
455 77 CHR(7)
456 RETURN
457
458 ELSE
459 CLEAR
460 SET DISPLAY TO VGA50
461
462
463 ENDIF
464
465 CASE reanetop = "VGA50"
466 IF ( ("VGA" & SYS( 206 ) )
467 77 CHR(7)
468 RETURN
469
470 ELSE
471 CLEAR
472 SET DISPLAY TO VGA50
473
474
475 ENDIF
476
477 CASE reanetop = "VGA50"
478 IF ( ("VGA" & SYS( 206 ) )
479 77 CHR(7)
480 RETURN
481
482 ELSE
483 CLEAR
484 SET DISPLAY TO VGA50
485
486
487 ENDIF
488
489 CASE reanetop = "VGA50"
490 IF ( ("VGA" & SYS( 206 ) )
491 77 CHR(7)
492 RETURN
493
494 ELSE
495 CLEAR
496 SET DISPLAY TO VGA50
497
498
499 ENDIF
500
501 CASE reanetop = "VGA50"
502 IF ( ("VGA" & SYS( 206 ) )
503 77 CHR(7)
504 RETURN
505
506 ELSE
507 CLEAR
508 SET DISPLAY TO VGA50
509
510
511 ENDIF
512
513 CASE reanetop = "VGA50"
514 IF ( ("VGA" & SYS( 206 ) )
515 77 CHR(7)
516 RETURN
517
518 ELSE
519 CLEAR
520 SET DISPLAY TO VGA50
521
522
523 ENDIF
524
525 CASE reanetop = "VGA50"
526 IF ( ("VGA" & SYS( 206 ) )
527 77 CHR(7)
528 RETURN
529
530 ELSE
531 CLEAR
532 SET DISPLAY TO VGA50
533
534
535 ENDIF
536
537 CASE reanetop = "VGA50"
538 IF ( ("VGA" & SYS( 206 ) )
539 77 CHR(7)
540 RETURN
541
542 ELSE
543 CLEAR
544 SET DISPLAY TO VGA50
545
546
547 ENDIF
548
549 CASE reanetop = "VGA50"
550 IF ( ("VGA" & SYS( 206 ) )
551 77 CHR(7)
552 RETURN
553
554 ELSE
555 CLEAR
556 SET DISPLAY TO VGA50
557
558
559 ENDIF
560
561 CASE reanetop = "VGA50"
562 IF ( ("VGA" & SYS( 206 ) )
563 77 CHR(7)
564 RETURN
565
566 ELSE
567 CLEAR
568 SET DISPLAY TO VGA50
569
570
571 ENDIF
572
573 CASE reanetop = "VGA50"
574 IF ( ("VGA" & SYS( 206 ) )
575 77 CHR(7)
576 RETURN
577
578 ELSE
579 CLEAR
580 SET DISPLAY TO VGA50
581
582
583 ENDIF
584
585 CASE reanetop = "VGA50"
586 IF ( ("VGA" & SYS( 206 ) )
587 77 CHR(7)
588 RETURN
589
590 ELSE
591 CLEAR
592 SET DISPLAY TO VGA50
593
594
595 ENDIF
596
597 CASE reanetop = "VGA50"
598 IF ( ("VGA" & SYS( 206 ) )
599 77 CHR(7)
600 RETURN
601
602 ELSE
603 CLEAR
604 SET DISPLAY TO VGA50
605
606
607 ENDIF
608
609 CASE reanetop = "VGA50"
610 IF ( ("VGA" & SYS( 206 ) )
611 77 CHR(7)
612 RETURN
613
614 ELSE
615 CLEAR
616 SET DISPLAY TO VGA50
617
618
619 ENDIF
620
621 CASE reanetop = "VGA50"
622 IF ( ("VGA" & SYS( 206 ) )
623 77 CHR(7)
624 RETURN
625
626 ELSE
627 CLEAR
628 SET DISPLAY TO VGA50
629
630
631 ENDIF
632
633 CASE reanetop = "VGA50"
634 IF ( ("VGA" & SYS( 206 ) )
635 77 CHR(7)
636 RETURN
637
638 ELSE
639 CLEAR
640 SET DISPLAY TO VGA50
641
642
643 ENDIF
644
645 CASE reanetop = "VGA50"
646 IF ( ("VGA" & SYS( 206 ) )
647 77 CHR(7)
648 RETURN
649
650 ELSE
651 CLEAR
652 SET DISPLAY TO VGA50
653
654
655 ENDIF
656
657 CASE reanetop = "VGA50"
658 IF ( ("VGA" & SYS( 206 ) )
659 77 CHR(7)
660 RETURN
661
662 ELSE
663 CLEAR
664 SET DISPLAY TO VGA50
665
666
667 ENDIF
668
669 CASE reanetop = "VGA50"
670 IF ( ("VGA" & SYS( 206 ) )
671 77 CHR(7)
672 RETURN
673
674 ELSE
675 CLEAR
676 SET DISPLAY TO VGA50
677
678
679 ENDIF
680
681 CASE reanetop = "VGA50"
682 IF ( ("VGA" & SYS( 206 ) )
683 77 CHR(7)
684 RETURN
685
686 ELSE
687 CLEAR
688 SET DISPLAY TO VGA50
689
690
691 ENDIF
692
693 CASE reanetop = "VGA50"
694 IF ( ("VGA" & SYS( 206 ) )
695 77 CHR(7)
696 RETURN
697
698 ELSE
699 CLEAR
700 SET DISPLAY TO VGA50
701
702
703 ENDIF
704
705 CASE reanetop = "VGA50"
706 IF ( ("VGA" & SYS( 206 ) )
707 77 CHR(7)
708 RETURN
709
710 ELSE
711 CLEAR
712 SET DISPLAY TO VGA50
713
714
715 ENDIF
716
717 CASE reanetop = "VGA50"
718 IF ( ("VGA" & SYS( 206 ) )
719 77 CHR(7)
720 RETURN
721
722 ELSE
723 CLEAR
724 SET DISPLAY TO VGA50
725
726
727 ENDIF
728
729 CASE reanetop = "VGA50"
730 IF ( ("VGA" & SYS( 206 ) )
731 77 CHR(7)
732 RETURN
733
734 ELSE
735 CLEAR
736 SET DISPLAY TO VGA50
737
738
739 ENDIF
740
741 CASE reanetop = "VGA50"
742 IF ( ("VGA" & SYS( 206 ) )
743 77 CHR(7)
744 RETURN
745
746 ELSE
747 CLEAR
748 SET DISPLAY TO VGA50
749
750
751 ENDIF
752
753 CASE reanetop = "VGA50"
754 IF ( ("VGA" & SYS( 206 ) )
755 77 CHR(7)
756 RETURN
757
758 ELSE
759 CLEAR
760 SET DISPLAY TO VGA50
761
762
763 ENDIF
764
765 CASE reanetop = "VGA50"
766 IF ( ("VGA" & SYS( 206 ) )
767 77 CHR(7)
768 RETURN
769
770 ELSE
771 CLEAR
772 SET DISPLAY TO VGA50
773
774
775 ENDIF
776
777 CASE reanetop = "VGA50"
778 IF ( ("VGA" & SYS( 206 ) )
779 77 CHR(7)
780 RETURN
781
782 ELSE
783 CLEAR
784 SET DISPLAY TO VGA50
785
786
787 ENDIF
788
789 CASE reanetop = "VGA50"
790 IF ( ("VGA" & SYS( 206 ) )
791 77 CHR(7)
792 RETURN
793
794 ELSE
795 CLEAR
796 SET DISPLAY TO VGA50
797
798
799 ENDIF
800
801 CASE reanetop = "VGA50"
802 IF ( ("VGA" & SYS( 206 ) )
803 77 CHR(7)
804 RETURN
805
806 ELSE
807 CLEAR
808 SET DISPLAY TO VGA50
809
810
811 ENDIF
812
813 CASE reanetop = "VGA50"
814 IF ( ("VGA" & SYS( 206 ) )
815 77 CHR(7)
816 RETURN
817
818 ELSE
819 CLEAR
820 SET DISPLAY TO VGA50
821
822
823 ENDIF
824
825 CASE reanetop = "VGA50"
826 IF ( ("VGA" & SYS( 206 ) )
827 77 CHR(7)
828 RETURN
829
830 ELSE
831 CLEAR
832 SET DISPLAY TO VGA50
833
834
835 ENDIF
836
837 CASE reanetop = "VGA50"
838 IF ( ("VGA" & SYS( 206 ) )
839 77 CHR(7)
840 RETURN
841
842 ELSE
843 CLEAR
844 SET DISPLAY TO VGA50
845
846
847 ENDIF
848
849 CASE reanetop = "VGA50"
850 IF ( ("VGA" & SYS( 206 ) )
851 77 CHR(7)
852 RETURN
853
854 ELSE
855 CLEAR
856 SET DISPLAY TO VGA50
857
858
859 ENDIF
860
861 CASE reanetop = "VGA50"
862 IF ( ("VGA" & SYS( 206 ) )
863 77 CHR(7)
864 RETURN
865
866 ELSE
867 CLEAR
868 SET DISPLAY TO VGA50
869
870
871 ENDIF
872
873 CASE reanetop = "VGA50"
874 IF ( ("VGA" & SYS( 206 ) )
875 77 CHR(7)
876 RETURN
877
878 ELSE
879 CLEAR
880 SET DISPLAY TO VGA50
881
882
883 ENDIF
884
885 CASE reanetop = "VGA50"
886 IF ( ("VGA" & SYS( 206 ) )
887 77 CHR(7)
888 RETURN
889
890 ELSE
891 CLEAR
892 SET DISPLAY TO VGA50
893
894
895 ENDIF
896
897 CASE reanetop = "VGA50"
898 IF ( ("VGA" & SYS( 206 ) )
899 77 CHR(7)
900 RETURN
901
902 ELSE
903 CLEAR
904 SET DISPLAY TO VGA50
905
906
907 ENDIF
908
909 CASE reanetop = "VGA50"
910 IF ( ("VGA" & SYS( 206 ) )
911 77 CHR(7)
912 RETURN
913
914 ELSE
915 CLEAR
916 SET DISPLAY TO VGA50
917
918
919 ENDIF
920
921 CASE reanetop = "VGA50"
922 IF ( ("VGA" & SYS( 206 ) )
923 77 CHR(7)
924 RETURN
925
926 ELSE
927 CLEAR
928 SET DISPLAY TO VGA50
929
930
931 ENDIF
932
933 CASE reanetop = "VGA50"
934 IF ( ("VGA" & SYS( 206 ) )
935 77 CHR(7)
936 RETURN
937
938 ELSE
939 CLEAR
940 SET DISPLAY TO VGA50
941
942
943 ENDIF
944
945 CASE reanetop = "VGA50"
946 IF ( ("VGA" & SYS( 206 ) )
947 77 CHR(7)
948 RETURN
949
950 ELSE
951 CLEAR
952 SET DISPLAY TO VGA50
953
954
955 ENDIF
956
957 CASE reanetop = "VGA50"
958 IF ( ("VGA" & SYS( 206 ) )
959 77 CHR(7)
960 RETURN
961
962 ELSE
963 CLEAR
964 SET DISPLAY TO VGA50
965
966
967 ENDIF
968
969 CASE reanetop = "VGA50"
970 IF ( ("VGA" & SYS( 206 ) )
971 77 CHR(7)
972 RETURN
973
974 ELSE
975 CLEAR
976 SET DISPLAY TO VGA50
977
978
979 ENDIF
980
981 CASE reanetop = "VGA50"
982 IF ( ("VGA" & SYS( 206 ) )
983 77 CHR(7)
984 RETURN
985
986 ELSE
987 CLEAR
988 SET DISPLAY TO VGA50
989
990
991 ENDIF
992
993 CASE reanetop = "VGA50"
994 IF ( ("VGA" & SYS( 206 ) )
995 77 CHR(7)
996 RETURN
997
998 ELSE
999 CLEAR
1000 SET DISPLAY TO VGA50
1001
1002
1003 ENDIF
1004
1005 CASE reanetop = "VGA50"
1006 IF ( ("VGA" & SYS( 206 ) )
1007 77 CHR(7)
1008 RETURN
1009
1010 ELSE
1011 CLEAR
1012 SET DISPLAY TO VGA50
1013
1014
1015 ENDIF
1016
1017 CASE reanetop = "VGA50"
1018 IF ( ("VGA" & SYS( 206 ) )
1019 77 CHR(7)
1020 RETURN
1021
1022 ELSE
1023 CLEAR
1024 SET DISPLAY TO VGA50
1025
1026
1027 ENDIF
1028
1029 CASE reanetop = "VGA50"
1030 IF ( ("VGA" & SYS( 206 ) )
1031 77 CHR(7)
1032 RETURN
1033
1034 ELSE
1035 CLEAR
1036 SET DISPLAY TO VGA50
1037
1038
1039 ENDIF
1040
1041 CASE reanetop = "VGA50"
1042 IF ( ("VGA" & SYS( 206 ) )
1043 77 CHR(7)
1044 RETURN
1045
1046 ELSE
1047 CLEAR
1048 SET DISPLAY TO VGA50
1049
1050
1051 ENDIF
1052
1053 CASE reanetop = "VGA50"
1054 IF ( ("VGA" & SYS( 206 ) )
1055 77 CHR(7)
1056 RETURN
1057
1058 ELSE
1059 CLEAR
1060 SET DISPLAY TO VGA50
1061
1062
1063 ENDIF
1064
1065 CASE reanetop = "VGA50"
1066 IF ( ("VGA" & SYS( 206 ) )
1067 77 CHR(7)
1068 RETURN
1069
1070 ELSE
1071 CLEAR
1072 SET DISPLAY TO VGA50
1073
1074
1075 ENDIF
1076
1077 CASE reanetop = "VGA50"
1078 IF ( ("VGA" & SYS( 206 ) )
1079 77 CHR(7)
1080 RETURN
1081
1082 ELSE
1083 CLEAR
1084 SET DISPLAY TO VGA50
1085
1086
1087 ENDIF
1088
1089 CASE reanetop = "VGA50"
1090 IF ( ("VGA" & SYS( 206 ) )
1091 77 CHR(7)
1092 RETURN
1093
1094 ELSE
1095 CLEAR
1096 SET DISPLAY TO VGA50
1097
1098
1099 ENDIF
1100
1101 CASE reanetop = "VGA50"
1102 IF ( ("VGA" & SYS( 206 ) )
1103 77 CHR(7)
1104 RETURN
1105
1106 ELSE
1107 CLEAR
1108 SET DISPLAY TO VGA50
1109
1110
1111 ENDIF
1112
1113 CASE reanetop = "VGA50"
1114 IF ( ("VGA" & SYS( 206 ) )
1115 77 CHR(7)
1116 RETURN
1117
1118 ELSE
1119 CLEAR
1120 SET DISPLAY TO VGA50
1121
1122
1123 ENDIF
1124
1125 CASE reanetop = "VGA50"
1126 IF ( ("VGA" & SYS( 206 ) )
1127 77 CHR(7)
1128 RETURN
1129
1130 ELSE
1131 CLEAR
1132 SET DISPLAY TO VGA50
1133
1134
1135 ENDIF
1136
1137 CASE reanetop = "VGA50"
1138 IF ( ("VGA" & SYS( 206 ) )
1139 77 CHR(7)
1140 RETURN
1141
1142 ELSE
1143 CLEAR
1144 SET DISPLAY TO VGA50
1145
1146
1147 ENDIF
1148
1149 CASE reanetop = "VGA50"
1150 IF ( ("VGA" & SYS( 206 ) )
1151 77 CHR(7)
1152 RETURN
1153
1154 ELSE
1155 CLEAR
1156 SET DISPLAY TO VGA50
1157
1158
1159 ENDIF
1160
1161 CASE reanetop = "VGA50"
1162 IF ( ("VGA" & SYS( 206 ) )
1163 77 CHR(7)
1164 RETURN
1165
1166 ELSE
1167 CLEAR
1168 SET DISPLAY TO VGA50
1169
1170
1171 ENDIF
1172
1173 CASE reanetop = "VGA50"
1174 IF ( ("VGA" & SYS( 206 ) )
1175 77 CHR(7)
1176 RETURN
1177
1178 ELSE
1179 CLEAR
1180 SET DISPLAY TO VGA50
1181
1182
1183 ENDIF
1184
1185 CASE reanetop = "VGA50"
1186 IF ( ("VGA" & SYS( 206 ) )
1187 77 CHR(7)
1188 RETURN
1189
1190 ELSE
1191 CLEAR
1192 SET DISPLAY TO VGA50
1193
1194
1195 ENDIF
1196
1197 CASE reanetop = "VGA50"
1198 IF ( ("VGA" & SYS( 206 ) )
1199 77 CHR(7)
1200 RETURN
1201
1202 ELSE
1203 CLEAR
1204 SET DISPLAY TO VGA50
1205
1206
1207 ENDIF
1208
1209 CASE reanetop = "VGA50"
1210 IF ( ("VGA" & SYS( 206 ) )
1211 77 CHR(7)
1212 RETURN
1213
1214 ELSE
1215 CLEAR
1216 SET DISPLAY TO VGA50
1217
1218
1219 ENDIF
1220
1221 CASE reanetop = "VGA50"
1222 IF ( ("VGA" & SYS( 206 ) )
1223 77 CHR(7)
1224 RETURN
1225
1226 ELSE
1227 CLEAR
1228 SET DISPLAY TO VGA50
1229
1230
1231 ENDIF
1232
1233 CASE reanetop = "VGA50"
1234 IF ( ("VGA" & SYS( 206 ) )
1235 77 CHR(7)
1236 RETURN
1237
1238 ELSE
1239 CLEAR
1240 SET DISPLAY TO VGA50
1241
1242
1243 ENDIF
1244
1245 CASE reanetop = "VGA50"
1246 IF ( ("VGA" & SYS( 206 ) )
1247 77 CHR(7)
1248 RETURN
1249
1250 ELSE
1251 CLEAR
1252 SET DISPLAY TO VGA50
1253
1254
1255 ENDIF
1256
1257 CASE reanetop = "VGA50"
1258 IF ( ("VGA" & SYS( 206 ) )
1259 77 CHR(7)
1260 RETURN
1261
1262 ELSE
1263 CLEAR
1264 SET DISPLAY TO VGA50
1265
1266
1267 ENDIF
1268
1269 CASE reanetop = "VGA50"
1270 IF ( ("VGA" & SYS( 206 ) )
1271 77 CHR(7)
1272 RETURN
1273
1274 ELSE
1275 CLEAR
1276 SET DISPLAY TO VGA50
1277
1278
1279 ENDIF
1280
1281 CASE reanetop = "VGA50"
1282 IF ( ("VGA" & SYS( 206 ) )
1283 77 CHR(7)
1284 RETURN
1285
1286 ELSE
1287 CLEAR
1288 SET DISPLAY TO VGA50
1289
1290
1291 ENDIF
1292
1293 CASE reanetop = "VGA50"
1294 IF ( ("VGA" & SYS( 206 ) )
1295 77 CHR(7)
1296 RETURN
1297
1298 ELSE
1299 CLEAR
1300 SET DISPLAY TO VGA50
1301
1302
1303 ENDIF
1304
1305 CASE reanetop = "VGA50"
1306 IF ( ("VGA" & SYS( 206 ) )
1307 77 CHR(7)
1308 RETURN
1309
1310 ELSE
1311 CLEAR
1312 SET DISPLAY TO VGA50
1313
1314
1315 ENDIF
1316
1317 CASE reanetop = "VGA50"
1318 IF ( ("VGA" & SYS( 206 ) )
1319 77 CHR(7)
1320 RETURN
1321
1322 ELSE
1323 CLEAR
1324 SET DISPLAY TO VGA50
1325
1326
1327 ENDIF
1328
1329 CASE reanetop = "VGA50"
1330 IF ( ("VGA" & SYS( 206 ) )
1331 77 CHR(7)
1332 RETURN
1333
1334 ELSE
1335 CLEAR
1336 SET DISPLAY TO VGA50
1337
1338
1339 ENDIF
1340
1341 CASE reanetop = "VGA50"
1342 IF ( ("VGA" & SYS( 206 ) )
1343 77 CHR(7)
1344 RETURN
1345
1346 ELSE
1347 CLEAR
1348 SET DISPLAY TO VGA50
1349
1350
1351 ENDIF
1352
1353 CASE reanetop = "VGA50"
1354 IF ( ("VGA" & SYS( 206 ) )
1355 77 CHR(7)
1356 RETURN
1357
1358 ELSE
1359 CLEAR
1360 SET DISPLAY TO VGA50
1361
1362
1363 ENDIF
1364
1365 CASE reanetop = "VGA50"
1366 IF ( ("VGA" & SYS( 206 ) )
1367 77 CHR(7)
1368 RETURN
1369
1370 ELSE
1371 CLEAR
1372 SET DISPLAY TO VGA50
1373
1374
1375 ENDIF
1376
1377 CASE reanetop = "VGA50"
1378 IF ( ("VGA" & SYS( 206 ) )
1379 77 CHR(7)
1380 RETURN
1381
1382 ELSE
1383 CLEAR
1384 SET DISPLAY TO VGA50
1385
1386
1387 ENDIF
1388
1389 CASE reanetop = "VGA50"
1390 IF ( ("VGA" & SYS( 206 ) )
1391 77 CHR(7)
1392 RETURN
1393
1394 ELSE
1395 CLEAR
1396 SET DISPLAY TO VGA50
1397
1398
1399 ENDIF
1400
1401 CASE reanetop = "VGA50"
1402 IF ( ("VGA" & SYS( 206 ) )
1403 77 CHR(7)
1404 RETURN
1405
1406 ELSE
1407 CLEAR
1408 SET DISPLAY TO VGA50
1409
1410
1411 ENDIF
1412
1413 CASE reanetop = "VGA50"
1414 IF ( ("VGA" & SYS( 206 ) )
1415 77 CHR(7)
1416 RETURN
1417
1418 ELSE
1419 CLEAR
1420 SET DISPLAY TO VGA50
1421
1422
1423 ENDIF
1424
1425 CASE reanetop = "VGA50"
1426 IF ( ("VGA" & SYS( 206 ) )
1427 77 CHR(7)
1428 RETURN
1429
1430 ELSE
1431 CLEAR
1432 SET DISPLAY TO VGA50
1433
1434
1435 ENDIF
1436
1437 CASE reanetop = "VGA50"
1438 IF ( ("VGA" & SYS( 206 ) )
1439 77 CHR(7)
1440 RETURN
1441
1442 ELSE
1443 CLEAR
1444 SET DISPLAY TO VGA50
1445
1446
1447 ENDIF
1448
1449 CASE reanetop = "VGA50"
1450 IF ( ("VGA" & SYS( 206 ) )
1451 77 CHR(7)
1452 RETURN
1453
1454 ELSE
1455 CLEAR
1456 SET DISPLAY TO VGA50
1457
1458
1459 ENDIF
1460
1461 CASE reanetop = "VGA50"
1462 IF ( ("VGA" & SYS( 206 ) )
1463 77 CHR(7)
1464 RETURN
1465
1466 ELSE
1467 CLEAR
1468 SET DISPLAY TO VGA50
1469
1470
1471 ENDIF
1472
1473 CASE reanetop = "VGA50"
1474 IF ( ("VGA" & SYS( 206 ) )
1475 77 CHR(7)
1476 RETURN
1477
1478 ELSE
1479 CLEAR
1480 SET DISPLAY TO VGA50
1481
1482
1483 ENDIF
1484
1485 CASE reanetop = "VGA50"
1486 IF ( ("VGA" & SYS( 206 ) )
1487 77 CHR(7)
1488 RETURN
1489
1490 ELSE
1491 CLEAR
1492 SET DISPLAY TO VGA50
1493
1494
1495 ENDIF
1496
1497 CASE reanetop = "VGA50"
1498 IF ( ("VGA" & SYS( 206 ) )
1499 77 CHR(7)
1500 RETURN
1501
1502 ELSE
1503 CLEAR
1504 SET DISPLAY TO VGA50
1505
1506
1507 ENDIF
1508
1509 CASE reanetop = "VGA50"
1510 IF ( ("VGA" & SYS( 206 ) )
1511 77 CHR(7)
1512 RETURN
1513
1514 ELSE
1515 CLEAR
1516 SET DISPLAY TO VGA50
1517
1518
1519 ENDIF
1520
1521 CASE reanetop = "VGA50"
1522 IF ( ("VGA" & SYS( 206 ) )
1523 77 CHR(7)
1524 RETURN
1525
1526 ELSE
1527 CLEAR
1528 SET DISPLAY TO VGA50
1529
1530
1531 ENDIF
1532
1533 CASE reanetop = "VGA50"
1534 IF ( ("VGA" & SYS( 206 ) )
1535 77 CHR(7)
1536 RETURN
1537
1538 ELSE
1539 CLEAR
1540 SET DISPLAY TO VGA50
1541
1542
1543 ENDIF
1544
1545 CASE reanetop = "VGA50"
1546 IF ( ("VGA" & SYS( 206 ) )
1547 77 CHR(7)
1548 RETURN
1549
1550 ELSE
1551 CLEAR
1552 SET DISPLAY TO VGA50
1553
1554
1555 ENDIF
1556
1557 CASE reanetop = "VGA50"
1558 IF ( ("VGA" & SYS( 206 ) )
1559 77 CHR(7)
1560 RETURN
1561
1562 ELSE
1563 CLEAR
1564 SET DISPLAY TO VGA50
1565
1566
1567 ENDIF
1568
1569 CASE reanetop = "VGA50"
1570 IF ( ("VGA" & SYS( 206 ) )
1571 77 CHR(7)
1572 RETURN
1573
1574 ELSE
1575 CLEAR
1576 SET DISPLAY TO VGA50
1577
1578
1579 ENDIF
1580
1581 CASE reanetop = "VGA50"
1582 IF ( ("VGA" & SYS( 206 ) )
1583 77 CHR(7)
1584 RETURN
1585
1586 ELSE
1587 CLEAR
1588 SET DISPLAY TO VGA50
1589
1590
1591 ENDIF
1592
1593 CASE reanetop = "VGA50"
1594 IF ( ("VGA" & SYS( 206 ) )
1595 77 CHR(7)
1596 RETURN
1597
1598 ELSE
1599 CLEAR
1600 SET DISPLAY TO VGA50
1601
1602
1603 ENDIF
1604
1605 CASE reanetop = "VGA50"
1606 IF ( ("VGA" & SYS( 206 ) )
1607 77 CHR(7)
1608 RETURN
1609
1610 ELSE
1611 CLEAR
1612 SET DISPLAY TO VGA50
1613
1614
1615 ENDIF
1616
1617 CASE reanetop = "VGA50"
1618 IF ( ("VGA" & SYS( 206 ) )
1619 77 CHR(7)
1620 RETURN
1621
1622 ELSE
1623 CLEAR
1624 SET DISPLAY TO VGA50
1625
1626
1627 ENDIF
1628
1629 CASE reanetop = "VGA50"
1630 IF ( ("VGA" & SYS( 206 ) )
1631 77 CHR(7)
1632 RETURN
1633
1634 ELSE
1635 CLEAR
1636 SET DISPLAY TO VGA50
1637
1638
1639 ENDIF
1640
1641 CASE reanetop = "VGA50"
1642 IF ( ("VGA" & SYS( 206 ) )
1643 77 CHR(7)
1644 RETURN
1645
1646 ELSE
1647 CLEAR
1648 SET DISPLAY TO VGA50
1649
1650
1651 ENDIF
1652
1653 CASE reanetop = "VGA50"
1654 IF ( ("VGA" & SYS( 206 ) )
1655 77 CHR(7)
1656 RETURN
1657
1658 ELSE
1659 CLEAR
1660 SET DISPLAY TO VGA50
1661
1662
1663 ENDIF
1664
1665 CASE reanetop = "VGA50"
1666 IF ( ("VGA" & SYS( 206 ) )
1667 77 CHR(7)
1668 RETURN
1669
1670 ELSE
1671 CLEAR
1672 SET DISPLAY TO VGA50
1673
1674
1675 ENDIF
1676
1677 CASE reanetop = "VGA50"
1678 IF ( ("VGA" & SYS( 206 ) )
1679 77 CHR(7)
1680 RETURN
1681
1682 ELSE
1683 CLEAR
1684 SET DISPLAY TO VGA50
1685
1686
1687 ENDIF
1688
1689 CASE reanetop = "VGA50"
1690 IF ( ("VGA" & SYS( 206 ) )
1691 77 CHR(7)
1692 RETURN
1693
1694 ELSE
1695 CLEAR
1696 SET DISPLAY TO VGA50
1697
1698
1699 ENDIF
1700
1701 CASE reanetop = "VGA50"
1702 IF ( ("VGA" & SYS( 206 ) )
1703 77 CHR(7)
1704 RETURN
1705
1706 ELSE
1707 CLEAR
1708 SET DISPLAY TO VGA50
1709
1710
1711 ENDIF
1712
1713 CASE reanetop = "VGA50"
1714 IF ( ("VGA" & SYS( 206 ) )
1715 77 CHR(7)
1716 RETURN
1717
1718 ELSE
1719 CLEAR
1720 SET DISPLAY TO VGA50
1721
172
```



```

528 ENDOF
529 RETURN
530 *****
531 ***** Procedure: SAYEOF
532 *****
533 *****
534 ***** (procedure in NPS_PROC.PRG)
535 ***** (procedure in NPS_PROC.PRG)
536 *****
537 *****
538 *****
539 *****
540 *****
541 *****
542 *****
543 *****
544 *****
545 *****
546 *****
547 *****
548 *****
549 *****
550 *****
551 *****
552 *****
553 *****
554 *****
555 *****
556 *****
557 *****
558 *****
559 *****
560 *****
561 *****
562 *****
563 *****
564 *****
565 *****
566 *****
567 *****
568 *****
569 *****
570 *****
571 *****
572 *****
573 *****
574 *****
575 *****
576 *****
577 *****
578 *****
579 *****
580 *****
581 *****
582 *****
583 *****
584 *****
585 *****
586 *****
587 *****
588 *****
589 *****
590 *****
591 *****
592 *****
593 *****
594 *****
595 *****
596 *****
597 *****
598 *****
599 *****
600 *****
601 *****
602 *****
603 *****
604 *****
605 *****
606 *****
607 *****
608 *****
609 *****
610 *****
611 *****
612 *****
613 *****
614 *****
615 *****

```

```

616 *****
617 *****
618 *****
619 *****
620 *****
621 *****
622 *****
623 *****
624 *****
625 *****
626 *****
627 *****
628 *****
629 *****
630 *****
631 *****
632 *****
633 *****
634 *****
635 *****
636 *****
637 *****
638 *****
639 *****
640 *****
641 *****
642 *****
643 *****
644 *****
645 *****
646 *****
647 *****
648 *****
649 *****
650 *****
651 *****
652 *****
653 *****
654 *****
655 *****
656 *****
657 *****
658 *****
659 *****
660 *****
661 *****
662 *****
663 *****
664 *****
665 *****
666 *****
667 *****
668 *****
669 *****
670 *****
671 *****
672 *****
673 *****
674 *****
675 *****
676 *****
677 *****
678 *****
679 *****
680 *****
681 *****
682 *****
683 *****
684 *****
685 *****
686 *****
687 *****
688 *****
689 *****
690 *****
691 *****
692 *****
693 *****
694 *****
695 *****
696 *****
697 *****
698 *****
699 *****
700 *****
701 *****
702 *****
703 *****

```

```

704 *****
705 *****
706 *****
707 *****
708 *****
709 *****
710 *****
711 *****
712 *****
713 *****
714 *****
715 *****
716 *****
717 *****
718 *****
719 *****
720 *****
721 *****
722 *****
723 *****
724 *****
725 *****
726 *****
727 *****
728 *****
729 *****
730 *****
731 *****
732 *****
733 *****
734 *****
735 *****
736 *****
737 *****
738 *****
739 *****
740 *****
741 *****
742 *****
743 *****
744 *****
745 *****
746 *****
747 *****
748 *****
749 *****
750 *****
751 *****
752 *****
753 *****
754 *****
755 *****
756 *****
757 *****
758 *****
759 *****
760 *****
761 *****
762 *****
763 *****
764 *****
765 *****
766 *****
767 *****
768 *****
769 *****
770 *****
771 *****
772 *****
773 *****
774 *****
775 *****
776 *****
777 *****
778 *****
779 *****
780 *****
781 *****
782 *****
783 *****
784 *****
785 *****
786 *****
787 *****
788 *****
789 *****
790 *****
791 *****

```



```

7972 PROCEDURE nps_seek
7973   PARAMETER ROW
7974   PRIVATE expir
7975   IF ndorder = "0"
7976     RETURN
7977   SET COLOR TO W/H,N/W
7978   DO CASE
7979     CASE defarea = "1"
7980       DO ROW D CLEAR
7981       CASE ndorder = "1"
7982         expir = SPACE(11)
7983         @ ROW, 0 SAY "Enter SSN" GET asssn PICTURE "999-99-9999"
7984         READ
7985         expir = TRIM(asssn)
7986         IF "" (> expir
7987           SEEK expir
7988         ENDF
7989       CASE ndorder = "2"
7990         alast_name = SPACE(23)
7991         first_name = SPACE(15)
7992         @ ROW, 0 SAY "Enter Last Name" GET alast_name ;
7993         PICTURE "AAAAAAAAAAAAAAA"
7994         @ ROW,1 D SAY "First Name" GET alfirst_name ;
7995         PICTURE "AAAAAAAAAAAAAAA"
7996         READ
7997         expir = TRIM(alast_name)+TRIM(alfirst_name)
7998         IF "" (> expir
7999           SEEK expir
8000         ENDF
8001       CASE defarea = "3"
8002         @ ROW D CLEAR
8003         expir = SPACE(3)
8004         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8005         READ
8006         expir = TRIM(acurrcur_nume)
8007         IF "" (> expir && If Ever not nul
8008           SEEK expir
8009         ENDF
8010       CASE defarea = "4"
8011         @ ROW D CLEAR
8012         expir = SPACE(15)
8013         @ ROW, 0 SAY "Enter UIC" GET mulcib PICTURE "01"
8014         READ
8015         expir = TRIM(mulcib)
8016         IF "" (> expir && If Ever not nul
8017           SEEK expir
8018         ENDF
8019       CASE defarea = "5"
8020         @ ROW D CLEAR
8021         expir = SPACE(3)
8022         @ ROW, 0 SAY "Enter UIC" GET mulcib PICTURE "01"
8023         READ
8024         expir = TRIM(mulcib)
8025         IF "" (> expir && If Ever not nul
8026           SEEK expir
8027         ENDF
8028       CASE defarea = "6"
8029         @ ROW D CLEAR
8030         expir = SPACE(15)
8031         @ ROW, 0 SAY "Enter UIC" GET mulcib PICTURE "01"
8032         READ
8033         expir = TRIM(mulcib)
8034         IF "" (> expir && If Ever not nul
8035           SEEK expir
8036         ENDF
8037       CASE defarea = "7"
8038         @ ROW D CLEAR
8039         expir = SPACE(3)
8040         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8041         READ
8042         expir = TRIM(acurrcur_nume)
8043         IF "" (> expir && If Ever not nul
8044           SEEK expir
8045         ENDF
8046       CASE defarea = "8"
8047         @ ROW D CLEAR
8048         expir = SPACE(3)
8049         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8050         READ
8051         expir = TRIM(acurrcur_nume)
8052         IF "" (> expir && If Ever not nul
8053           SEEK expir
8054         ENDF
8055       CASE defarea = "9"
8056         @ ROW D CLEAR
8057         expir = SPACE(3)
8058         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8059         READ
8060         expir = TRIM(acurrcur_nume)
8061         IF "" (> expir && If Ever not nul
8062           SEEK expir
8063         ENDF
8064       CASE defarea = "A"
8065         @ ROW D CLEAR
8066         expir = SPACE(3)
8067         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8068         READ
8069         expir = TRIM(acurrcur_nume)
8070         IF "" (> expir && If Ever not nul
8071           SEEK expir
8072         ENDF
8073       CASE defarea = "B"
8074         @ ROW D CLEAR
8075         expir = SPACE(3)
8076         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8077         READ
8078         expir = TRIM(acurrcur_nume)
8079         IF "" (> expir && If Ever not nul
8080           SEEK expir
8081         ENDF
8082       CASE defarea = "C"
8083         @ ROW D CLEAR
8084         expir = SPACE(3)
8085         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8086         READ
8087         expir = TRIM(acurrcur_nume)
8088         IF "" (> expir && If Ever not nul
8089           SEEK expir
8090         ENDF
8091       CASE defarea = "D"
8092         @ ROW D CLEAR
8093         expir = SPACE(3)
8094         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8095         READ
8096         expir = TRIM(acurrcur_nume)
8097         IF "" (> expir && If Ever not nul
8098           SEEK expir
8099         ENDF
8100       CASE defarea = "E"
8101         @ ROW D CLEAR
8102         expir = SPACE(3)
8103         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8104         READ
8105         expir = TRIM(acurrcur_nume)
8106         IF "" (> expir && If Ever not nul
8107           SEEK expir
8108         ENDF
8109       CASE defarea = "F"
8110         @ ROW D CLEAR
8111         expir = SPACE(3)
8112         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8113         READ
8114         expir = TRIM(acurrcur_nume)
8115         IF "" (> expir && If Ever not nul
8116           SEEK expir
8117         ENDF
8118       CASE defarea = "G"
8119         @ ROW D CLEAR
8120         expir = SPACE(3)
8121         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8122         READ
8123         expir = TRIM(acurrcur_nume)
8124         IF "" (> expir && If Ever not nul
8125           SEEK expir
8126         ENDF
8127       CASE defarea = "H"
8128         @ ROW D CLEAR
8129         expir = SPACE(3)
8130         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8131         READ
8132         expir = TRIM(acurrcur_nume)
8133         IF "" (> expir && If Ever not nul
8134           SEEK expir
8135         ENDF
8136       CASE defarea = "I"
8137         @ ROW D CLEAR
8138         expir = SPACE(3)
8139         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8140         READ
8141         expir = TRIM(acurrcur_nume)
8142         IF "" (> expir && If Ever not nul
8143           SEEK expir
8144         ENDF
8145       CASE defarea = "J"
8146         @ ROW D CLEAR
8147         expir = SPACE(3)
8148         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8149         READ
8150         expir = TRIM(acurrcur_nume)
8151         IF "" (> expir && If Ever not nul
8152           SEEK expir
8153         ENDF
8154       CASE defarea = "K"
8155         @ ROW D CLEAR
8156         expir = SPACE(3)
8157         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8158         READ
8159         expir = TRIM(acurrcur_nume)
8160         IF "" (> expir && If Ever not nul
8161           SEEK expir
8162         ENDF
8163       CASE defarea = "L"
8164         @ ROW D CLEAR
8165         expir = SPACE(3)
8166         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8167         READ
8168         expir = TRIM(acurrcur_nume)
8169         IF "" (> expir && If Ever not nul
8170           SEEK expir
8171         ENDF
8172       CASE defarea = "M"
8173         @ ROW D CLEAR
8174         expir = SPACE(3)
8175         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8176         READ
8177         expir = TRIM(acurrcur_nume)
8178         IF "" (> expir && If Ever not nul
8179           SEEK expir
8180         ENDF
8181       CASE defarea = "N"
8182         @ ROW D CLEAR
8183         expir = SPACE(3)
8184         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8185         READ
8186         expir = TRIM(acurrcur_nume)
8187         IF "" (> expir && If Ever not nul
8188           SEEK expir
8189         ENDF
8190       CASE defarea = "O"
8191         @ ROW D CLEAR
8192         expir = SPACE(3)
8193         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8194         READ
8195         expir = TRIM(acurrcur_nume)
8196         IF "" (> expir && If Ever not nul
8197           SEEK expir
8198         ENDF
8199       CASE defarea = "P"
8200         @ ROW D CLEAR
8201         expir = SPACE(3)
8202         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8203         READ
8204         expir = TRIM(acurrcur_nume)
8205         IF "" (> expir && If Ever not nul
8206           SEEK expir
8207         ENDF
8208       CASE defarea = "Q"
8209         @ ROW D CLEAR
8210         expir = SPACE(3)
8211         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8212         READ
8213         expir = TRIM(acurrcur_nume)
8214         IF "" (> expir && If Ever not nul
8215           SEEK expir
8216         ENDF
8217       CASE defarea = "R"
8218         @ ROW D CLEAR
8219         expir = SPACE(3)
8220         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8221         READ
8222         expir = TRIM(acurrcur_nume)
8223         IF "" (> expir && If Ever not nul
8224           SEEK expir
8225         ENDF
8226       CASE defarea = "S"
8227         @ ROW D CLEAR
8228         expir = SPACE(3)
8229         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8230         READ
8231         expir = TRIM(acurrcur_nume)
8232         IF "" (> expir && If Ever not nul
8233           SEEK expir
8234         ENDF
8235       CASE defarea = "T"
8236         @ ROW D CLEAR
8237         expir = SPACE(3)
8238         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8239         READ
8240         expir = TRIM(acurrcur_nume)
8241         IF "" (> expir && If Ever not nul
8242           SEEK expir
8243         ENDF
8244       CASE defarea = "U"
8245         @ ROW D CLEAR
8246         expir = SPACE(3)
8247         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"
8248         READ
8249         expir = TRIM(acurrcur_nume)
8250         IF "" (> expir && If Ever not nul
8251           SEEK expir
8252         ENDF
8253       CASE defarea = "V"
8254         @ ROW D CLEAR
8255         expir = SPACE(3)
8256         @ ROW, 0 SAY "Enter Curriculum # " GET acurrcur_nume PICTURE "999-"

```

```

1144 STORE activity.act_name TO act_name
1145 STORE activity.acronym TO acronym
1146 STORE activity.poc TO poc
1147 RETURN
1148 RETURN
1149 IF
1150 IF
1151 IF
1152 IF
1153 IF
1154 IF
1155 IF
1156 IF
1157 IF
1158 IF
1159 IF
1160 IF
1161 IF
1162 IF
1163 IF
1164 IF
1165 IF
1166 IF
1167 IF
1168 IF
1169 IF
1170 IF
1171 IF
1172 IF
1173 IF
1174 IF
1175 IF
1176 IF
1177 IF
1178 IF
1179 IF
1180 IF
1181 IF
1182 IF
1183 IF
1184 IF
1185 IF
1186 IF
1187 IF
1188 IF
1189 IF
1190 IF
1191 IF
1192 IF
1193 IF
1194 IF
1195 IF
1196 IF
1197 IF
1198 IF
1199 IF
1200 IF
1201 IF
1202 IF
1203 IF
1204 IF
1205 IF
1206 IF
1207 IF
1208 IF
1209 IF
1210 IF
1211 IF
1212 IF
1213 IF
1214 IF
1215 IF
1216 IF
1217 IF
1218 IF
1219 IF
1220 IF
1221 IF
1222 IF
1223 IF
1224 IF
1225 IF
1226 IF
1227 IF
1228 IF
1229 IF
1230 IF
1231 IF
1232 IF
1233 IF
1234 IF
1235 IF
1236 IF
1237 IF
1238 IF
1239 IF
1240 IF
1241 IF
1242 IF
1243 IF
1244 IF
1245 IF
1246 IF
1247 IF
1248 IF
1249 IF
1250 IF
1251 IF
1252 IF
1253 IF
1254 IF
1255 IF
1256 IF
1257 IF
1258 IF
1259 IF
1260 IF
1261 IF
1262 IF
1263 IF
1264 IF
1265 IF
1266 IF
1267 IF
1268 IF
1269 IF
1270 IF
1271 IF
1272 IF
1273 IF
1274 IF
1275 IF
1276 IF
1277 IF
1278 IF
1279 IF
1280 IF
1281 IF
1282 IF
1283 IF
1284 IF
1285 IF
1286 IF
1287 IF
1288 IF
1289 IF
1290 IF
1291 IF
1292 IF
1293 IF
1294 IF
1295 IF
1296 IF
1297 IF
1298 IF
1299 IF
1300 IF
1301 IF
1302 IF
1303 IF
1304 IF
1305 IF
1306 IF
1307 IF
1308 IF
1309 IF
1310 IF
1311 IF
1312 IF
1313 IF
1314 IF
1315 IF
1316 IF
1317 IF

```

```

1054 IF
1055 IF
1056 IF
1057 IF
1058 IF
1059 IF
1060 IF
1061 IF
1062 IF
1063 IF
1064 IF
1065 IF
1066 IF
1067 IF
1068 IF
1069 IF
1070 IF
1071 IF
1072 IF
1073 IF
1074 IF
1075 IF
1076 IF
1077 IF
1078 IF
1079 IF
1080 IF
1081 IF
1082 IF
1083 IF
1084 IF
1085 IF
1086 IF
1087 IF
1088 IF
1089 IF
1090 IF
1091 IF
1092 IF
1093 IF
1094 IF
1095 IF
1096 IF
1097 IF
1098 IF
1099 IF
1100 IF
1101 IF
1102 IF
1103 IF
1104 IF
1105 IF
1106 IF
1107 IF
1108 IF
1109 IF
1110 IF
1111 IF
1112 IF
1113 IF
1114 IF
1115 IF
1116 IF
1117 IF
1118 IF
1119 IF
1120 IF
1121 IF
1122 IF
1123 IF
1124 IF
1125 IF
1126 IF
1127 IF
1128 IF
1129 IF
1130 IF
1131 IF
1132 IF
1133 IF
1134 IF
1135 IF
1136 IF
1137 IF
1138 IF
1139 IF
1140 IF
1141 IF
1142 IF
1143 IF

```


[illegible]

```

1575      ENDIF
1576      ENDIF
1577      RETURN wrk_page
1578      *****
1579      *****
1580      Procedure: PRNCHK
1581      *****
1582      Called by: NPS.RECA.PRG
1583      ; REC_LISTS
1584      ; PRNPR
1585      ;
1586      ; Calls: GETKEY
1587      ;
1588      *****
1589      Procedure: prnchk
1590      *****
1591      SET COLOR TO W/N,N/W
1592      @ 7,24 TO 14,55 COLOR W/RB
1593      @ 8,25 SAY " - Check the printer ..."
1594      @ 9,25 SAY " - Is the power on?"
1595      @ 10,25 SAY " - Is it on-line?"
1596      @ 11,25 SAY " - Is it set to 12-pitch?"
1597      @ 12,25 SAY " - Is there enough paper?"
1598      @ 13,25 SAY " - Is the ribbon OK?"
1599      SET CURSOR ON
1600      @ 16,25 SAY "Continue with print job? (y/n)"
1601      choice = ""
1602      DO GETKEY WITH choice, "YN"
1603      prnstat = choice
1604      RETURN
1605      *****
1606      *****
1607      Procedure: PRNSTOP
1608      *****
1609      Called by: NPS.RECA.PRG
1610      ; RECPRN
1611      ; PRNPR
1612      ;
1613      *****
1614      Procedure: prnstop
1615      *****
1616      @ 10,10 TO 11,79 CLEAR
1617      IF aprint = .T.
1618      STORE .F. TO aprint
1619      EJECT
1620      WAIT "Print job aborted." WINDOW TIMEOUT 2
1621      ENDIF
1622      RETURN
1623      *****
1624      *****
1625      Procedure: PRNPROB
1626      *****
1627      Called by: NPS.RECA.PRG
1628      ; REC_LISTS
1629      ; RECPRN
1630      ; PRNPR
1631      *****
1632      Procedure: prnprob
1633      *****
1634      aprint = .F.
1635      IF ERROR() = 125
1636      77 CHR(7)
1637      WAIT "Printer problem. Check the printer." WINDOW TIMEOUT 1
1638      ENDIF
1639      RETURN
1640      *****
1641      *****
1642      Procedure: SETPRN
1643      *****
1644      Called by: GORECALLS
1645      ; PRNPR
1646      ;
1647      *****
1648      Procedure: setprn
1649      *****
1650      IF nps.elec.printer = "Ign Proprinter"
1651      ELSE 77 CHR(27) + CHR(58)
1652      ELSE 77 CHR(27) + "n"
1653      ENDIF
1654      RETURN
1655      *****

```


INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2. Library, Code 52 Naval Postgraduate School Monterey, California 93943-5002	2
3. LCDR Timothy P. Steele, MSC, USN 12339 Coleraine Court Reston, Virginia 22091	2
4. Prof. Hemant K. Bhargava, Code AS/BH Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5000	1
5. Prof. Dani Zweig, Code AS/ZG Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5000	1
6. Director, Branch Dental Clinic Naval Postgraduate School Monterey, California 93943-5100	1
7. Prof. Tung X. Bui Code AS/BD Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5000	1



3 2768 00036904 5